aws

Leitfaden

Amazon CloudWatch

Amazon CloudWatch: Leitfaden

Table of Contents

Was ist Amazon CloudWatch?

Amazon CloudWatch überwacht Ihre Amazon Web Services (AWS) -Ressourcen und die Anwendungen, auf denen Sie laufen, AWS in Echtzeit. Sie können CloudWatch damit Metriken sammeln und verfolgen. Dabei handelt es sich um Variablen, die Sie für Ihre Ressourcen und Anwendungen messen können.

Auf der CloudWatch Startseite werden automatisch Metriken zu jedem AWS Service angezeigt, den Sie verwenden. Sie können zusätzlich benutzerdefinierte Dashboards erstellen, um Metriken über Ihre benutzerdefinierten Anwendungen anzuzeigen und benutzerdefinierte Sammlungen von Metriken Ihrer Wahl anzuzeigen.

Sie können Alarme erstellen, die Metriken überwachen und Benachrichtigungen senden oder automatisch Änderungen an den Ressourcen vornehmen, die Sie überwachen, wenn ein Schwellenwert überschritten wird. Beispielsweise können Sie die CPU-Auslastung und Datenträgerlese- und -schreibvorgänge Ihrer Amazon-EC2-Instances überwachen lassen und diese Daten dann verwenden, um festzustellen, ob Sie zusätzliche Instances starten sollten, um die erhöhte Last zu bewältigen. Sie können diese Daten auch verwenden, um unausgelastete Instances anzuhalten und dadurch Geld zu sparen.

Mit CloudWatch erhalten Sie systemweiten Einblick in die Ressourcennutzung, die Anwendungsleistung und den Betriebsstatus.

Zugreifen CloudWatch

Sie können CloudWatch mit einer der folgenden Methoden darauf zugreifen:

- CloudWatch Amazon-Konsole — https://console.aws.amazon.com/cloudwatch/
- AWS CLI — Weitere Informationen finden Sie unter Getting Setup with the AWS Command Line Interface im AWS Command Line Interface Benutzerhandbuch.
- CloudWatch API — Weitere Informationen finden Sie in der Amazon CloudWatch API-Referenz.
- AWS SDKs — Weitere Informationen finden Sie unter Tools für Amazon Web Services.

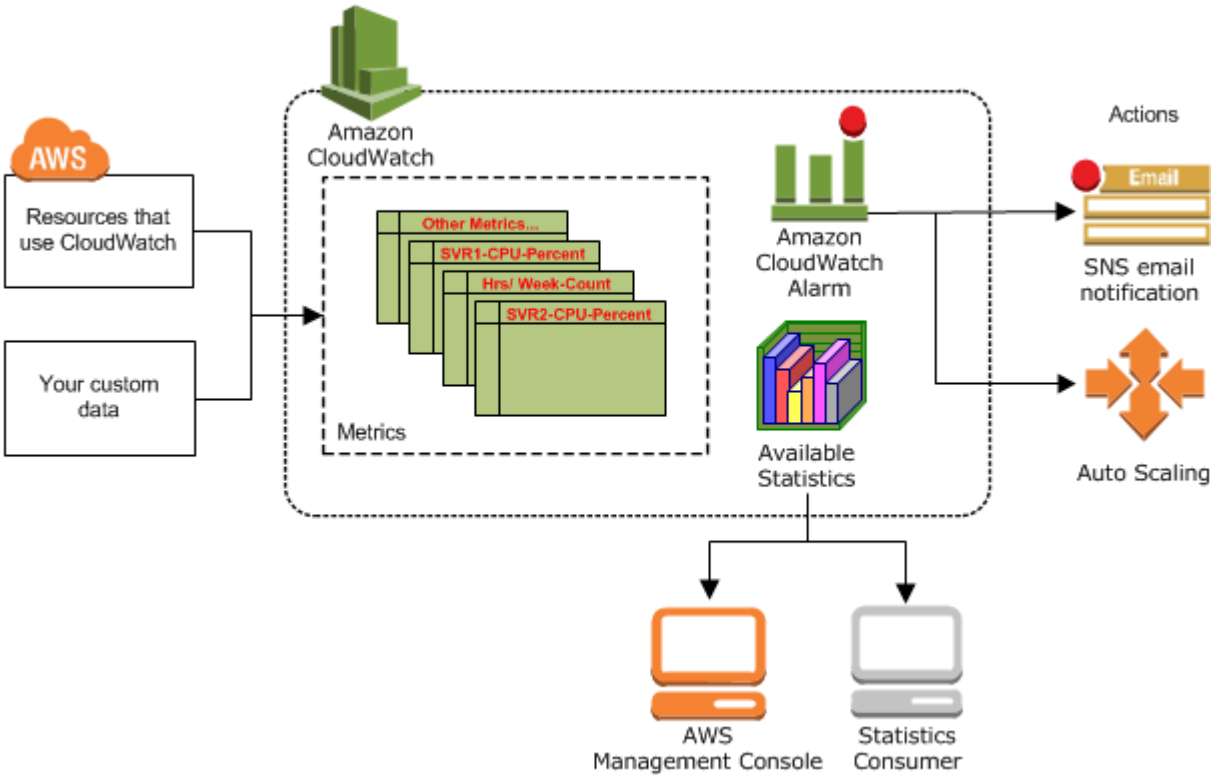
Verwandte Dienste AWS

Die folgenden Dienste werden zusammen mit Amazon genutzt CloudWatch:

- Amazon Simple Notification Service (Amazon SNS) koordiniert und verwalte die Zustellung oder das Senden von Nachrichten an abonnierende Endpunkte oder Clients. Sie verwenden Amazon SNS withCloudWatch , um Nachrichten zu senden, wenn ein Alarmschwellenwert erreicht wurde. Weitere Informationen finden Sie unter Einrichten von Amazon-SNS-Benachrichtigungen.
- Mit Amazon EC2 Auto Scaling können Sie Amazon-EC2-Instances basierend auf benutzerdefinierten Richtlinien, Zustandsüberprüfungen und Plänen automatisch starten oder beenden. Sie können einen CloudWatch Alarm mit Amazon EC2 Auto Scaling verwenden, um Ihre EC2-Instances je nach Bedarf zu skalieren. Weitere Informationen finden Sie unter Dynamische Skalierung im Benutzerhandbuch für Amazon EC2 Auto Scaling.
- AWS CloudTrailermöglicht es Ihnen, die Aufrufe der CloudWatch Amazon-API für Ihr Konto zu überwachen, einschließlich der Aufrufe von AWS Management Console AWS CLI, und anderen Diensten. Wenn die CloudTrail Protokollierung aktiviert ist, werden Protokolldateien in den Amazon S3 S3-Bucket CloudWatch geschrieben, den Sie bei der Konfiguration angegeben haben CloudTrail. Weitere Informationen finden Sie unter Protokollierung Amazon CloudWatch Amazon-API-Aufrufen mit AWS CloudTrail.
- AWS Identity and Access Management (IAM) ist ein Webservice, mit dem Sie den Zugriff Ihrer Benutzer auf AWS Ressourcen sicher kontrollieren können. Kontrollieren Sie mit IAM, wer Ihre AWS -Ressourcen verwenden kann (Authentifizierung) und welche Ressourcen auf welche Weise verwendet werden können (Autorisierung). Weitere Informationen finden Sie unter Identitäts- und Zugriffsmanagement für Amazon CloudWatch.

So CloudWatch funktioniert Amazon

Amazon CloudWatch ist im Grunde ein Metrik-Repository. Ein AWS Service — wie Amazon EC2 — speichert Metriken in das Repository, und Sie rufen Statistiken auf der Grundlage dieser Metriken ab. Wenn Sie Ihren eigenen benutzerdefinierten Metriken im Repository platzieren, können Sie ebenfalls auf diesen Metriken beruhende Statistiken abrufen.



Sie können Metriken verwenden, um Statistiken zu berechnen und die Daten dann grafisch in der Konsole darzustellen. CloudWatch Weitere Informationen zu den anderen AWS Ressourcen, die Metriken generieren und an die sie senden CloudWatch, finden Sie unterAWS Dienste, die CloudWatch Metriken veröffentlichen.

Sie können Alarmaktionen konfigurieren, um eine Amazon-EC2-Instance anzuhalten, zu starten oder zu beenden, wenn bestimmte Kriterien erfüllt sind. Darüber hinaus können Sie Alarme erstellen, die in Ihrem Namen Aktionen von Amazon EC2 Auto Scaling und Amazon Simple Notification Service (Amazon SNS) auslösen. Weitere Informationen zum Erstellen von CloudWatch Alarmen finden Sie unterAlarme.

AWS Cloud-Computing-Ressourcen sind in hochverfügbaren Rechenzentrumseinrichtungen untergebracht. Um eine zusätzliche Skalierbarkeit und Zuverlässigkeit zu bieten, befinden sich alle Rechenzentrumsanlagen in einem bestimmten geografischen Bereich, der auch als Region bezeichnet wird. Um eine größtmögliche Fehlerisolierung und Stabilität zu erreichen, ist jede Region so ausgelegt, dass sie vollständig von den anderen Regionen isoliert ist. Metriken werden separat in Regionen gespeichert, aber Sie können die CloudWatch regionsübergreifende Funktionalität verwenden, um Statistiken aus verschiedenen Regionen zu aggregieren. Weitere Informationen finden Sie unter Kontoübergreifende, regionsübergreifende Konsole CloudWatch und Regionen und Endpunkte im Allgemeine Amazon Web Services-Referenz .

CloudWatch Amazon-Konzepte

Die folgenden Begriffe und Konzepte sind für Ihr Verständnis und Ihre Nutzung von Amazon von zentraler Bedeutung CloudWatch:

- Namespaces
- Metriken
- Dimensionen
- Auflösung
- Statistiken
- Perzentile
- Alarme

Informationen zu den Servicekontingenten für CloudWatch Messwerte, Alarme, API-Anfragen und Alarm-E-Mail-Benachrichtigungen finden Sie unter CloudWatch Servicekontingenten.

Namespaces

Ein Namespace ist ein Container für CloudWatch Metriken. Metriken in verschiedenen Namespaces sind voneinander isoliert, damit Metriken aus unterschiedlichen Anwendungen nicht versehentlich in dieselben Statistiken aggregiert werden.

Es gibt keinen Standard-Namespace. Sie müssen für jeden Datenpunkt, in dem Sie veröffentlichen, einen Namespace angeben. CloudWatch Sie können beim Erstellen einer Metrik einen Namespace-Namen festlegen. Diese Namen müssen gültige ASCII-Zeichen enthalten und 255 oder weniger Zeichen lang sein. Mögliche Zeichen sind: alphanumerische Zeichen (0-9a-Za-Z), Punkt (.), Bindestrich (-), Unterstrich (_), Schrägstrich (/), Hash (#), Doppelpunkt (:) und das Leerzeichen. Ein Namespace muss mindestens ein Zeichen enthalten, das kein Leerzeichen ist.

Die AWS Namespaces verwenden in der Regel die folgende Namenskonvention:. `AWS/service` Amazon EC2 verwendet beispielsweise den Namespace `AWS/EC2`. Eine Liste der AWS Namespaces finden Sie unter. AWS Dienste, die CloudWatch Metriken veröffentlichen

Metriken

Metriken sind das grundlegende Konzept von. CloudWatch Eine Metrik stellt einen zeitlich geordneten Satz von Datenpunkten dar, die veröffentlicht werden. CloudWatch Sie können sich eine

Metrik als eine zu überwachende Variable und die Datenpunkte als die Werte dieser Variablen im Laufe der Zeit vorstellen. Die CPU-Auslastung einer bestimmten EC2-Instance ist beispielsweise eine von Amazon EC2 bereitgestellte Metrik. Die Datenpunkte selbst können aus einer beliebigen Anwendung oder geschäftlichen Aktivität stammen, aus der Sie Daten erheben.

Standardmäßig bieten viele AWS Services kostenlos Metriken für Ressourcen (wie Amazon EC2 EC2-Instances, Amazon EBS-Volumes und Amazon RDS-DB-Instances). Gegen eine Gebühr können Sie auch eine detaillierte Überwachung für einige Ressourcen wie Ihre Amazon-EC2-Instances aktivieren oder Ihre eigenen Anwendungsmetriken veröffentlichen. Für benutzerdefinierte Metriken können Sie die Datenpunkte in beliebiger Reihenfolge und beliebiger Geschwindigkeit hinzufügen. Sie können Statistiken zu diesen Datenpunkten als eine geordnete Reihe von Zeitreihendaten abrufen.

Metriken existieren nur in der Region, in der sie erstellt wurden. Metriken können zwar nicht gelöscht werden, laufen aber nach 15 Monaten ab, wenn in ihnen keine neuen Daten veröffentlicht werden. Datenpunkte, die älter als 15 Monate sind, laufen auf fortlaufender Basis ab, sobald neue Datenpunkte eintreffen. Daten, die älter als 15 Monate sind, werden verworfen.

Metriken werden eindeutig durch einen Namen, ein Namespace und keine oder mehrere Dimensionen definiert. Jeder Datenpunkt in einer Metrik verfügt über einen Zeitstempel und (optional) über eine Maßeinheit. Sie können Statistiken CloudWatch für jede Metrik abrufen.

Weitere Informationen finden Sie unter Anzeigen der verfügbaren Metriken und Veröffentlichen von benutzerdefinierten -Metriken.

Zeitstempel

Jeder Metrik-Datenpunkt muss einem Zeitstempel zugeordnet sein. Der Zeitstempel kann bis zu zwei Wochen in der Vergangenheit und bis zu zwei Stunden in der Zukunft liegen. Wenn Sie keinen Zeitstempel angeben, CloudWatch erstellt es einen Zeitstempel für Sie, der auf der Zeit basiert, zu der der Datenpunkt empfangen wurde.

Zeitstempel sind `dateTime` Objekte mit dem vollständigen Datum plus Stunden, Minuten und Sekunden (z. B. 2016-10-31T23:59:59Z). Weitere Informationen finden Sie unter dateTime.
Auch wenn dies nicht erforderlich ist, empfehlen wir die Verwendung der koordinierten Weltzeit (Coordinated Universal Time, UTC). Wenn Sie Statistiken von abrufen CloudWatch, werden alle Zeiten in UTC angegeben.

CloudWatch Alarme überprüfen die Messwerte auf der Grundlage der aktuellen Uhrzeit in UTC. Benutzerdefinierte Messwerte, an die CloudWatch mit anderen Zeitstempeln als der aktuellen UTC-

Zeit gesendet werden, können dazu führen, dass bei Alarmen der Status „Ungenügend Daten“ angezeigt wird, oder dass Alarme verzögert werden.

Speicherung von Metriken

CloudWatch speichert metrische Daten wie folgt:

- Datenpunkte mit einem Zeitraum von weniger als 60 Sekunden stehen 3 Stunden lang zur Verfügung. Bei diesen Datenpunkten handelt es sich um hochauflösende, benutzerdefinierte Metriken.
- Datenpunkte mit einem Zeitraum von 60 Sekunden (1 Minute) stehen 15 Tage lang zur Verfügung
- Datenpunkte mit einem Zeitraum von 300 Sekunden (5 Minuten) stehen 63 Tage lang zur Verfügung
- Datenpunkte mit einem Zeitraum von 3.600 Sekunden (1 Stunde) stehen für 455 Tage (15 Monate) zur Verfügung

Datenpunkte, die ursprünglich mit einem kürzeren Zeitraum veröffentlicht wurden, werden für eine langfristige Speicherung aggregiert. Wenn Sie z. B. Daten mit einem Zeitraum von 1 Minute sammeln, bleiben die Daten für 15 Tage mit einer Auflösung von 1 Minute verfügbar. Nach 15 Tagen sind die Daten noch immer verfügbar, aber sie sind aggregiert und können nur mit einer Auflösung von 5 Minuten abgerufen werden. Nach 63 Tagen werden die Daten weiter aggregiert und sind nur mit einer Auflösung von 1 Stunde verfügbar.

> **Note**
>
> Metriken, für die in den letzten zwei Wochen keine neuen Datenpunkte vorlagen, werden nicht in der Konsole angezeigt. Sie werden auch nicht angezeigt, wenn Sie den Metriknamen oder die Dimensionsnamen in der Konsole in das Suchfeld auf der Registerkarte All metrics (Alle Metriken) eingeben, und sie werden nicht in den Ergebnissen eines Befehls vom Typ list-metrics zurückgegeben. Diese Metriken lassen sich am besten mit den get-metric-statisticsBefehlen get-metric-dataoder in der abrufen AWS CLI.

Dimensionen

Eine Dimension ist ein Name-Wert-Paar, das zur Identifizierung einer Metrik beiträgt. Sie können einer Metrik bis zu 30 Dimensionen zuweisen.

Jede Metrik besitzt spezifische Eigenschaften, die sie beschreiben. Sie können sich Dimensionen als Kategorien für diese Eigenschaften vorstellen. Dimensionen unterstützen Sie dabei, eine Struktur für Ihren Statistikplan zu planen. Da Dimensionen Teil der eindeutigen ID für eine Metrik sind, erstellen Sie eine neue Variation dieser Metrik, sobald Sie einer Ihrer Metriken ein eindeutiges Namen-Werte-Paar hinzufügen.

AWS Dienste, die Daten senden, um jeder Metrik Dimensionen CloudWatch zuzuordnen. Sie können Dimensionen verwenden, um die zurückgegebenen Ergebnisse zu filtern. CloudWatch Sie können beispielsweise Statistiken für eine bestimmte EC2-Instance abrufen, indem Sie bei der Suche nach Metriken die Dimension `InstanceId` angeben.

Denn Metriken, die von bestimmten AWS Diensten wie Amazon EC2 erstellt wurden, CloudWatch können Daten dimensionsübergreifend aggregieren. Wenn Sie beispielsweise im `AWS/EC2` Namespace nach Metriken suchen, aber keine Dimensionen angeben, werden alle Daten für die angegebene Metrik CloudWatch zusammengefasst, um die von Ihnen angeforderte Statistik zu erstellen. CloudWatch aggregiert Ihre benutzerdefinierten Metriken nicht dimensionsübergreifend.

Kombinationen von Dimensionen

CloudWatch behandelt jede eindeutige Kombination von Dimensionen als separate Metrik, auch wenn die Metriken denselben Metriknamen haben. Sie können nur Statistiken abrufen, die Kombinationen von Dimensionen verwenden, die Sie speziell veröffentlicht haben. Wenn Sie Statistiken abrufen, geben Sie dieselben Werte für den Namespace, Metriknamen und die Dimensionsparameter an, die auch bei der Erstellung der Metriken verwendet wurden. Sie können auch die Start- und Endzeiten angeben, die für CloudWatch die Aggregation verwendet werden sollen.

Nehmen wir beispielsweise an, Sie veröffentlichen vier verschiedene Metriken, die ServerStats im DataCenterMetric Namespace benannt sind und die folgenden Eigenschaften haben:

```
Dimensions: Server=Prod, Domain=Frankfurt, Unit: Count, Timestamp:
 2016-10-31T12:30:00Z, Value: 105
Dimensions: Server=Beta, Domain=Frankfurt, Unit: Count, Timestamp:
 2016-10-31T12:31:00Z, Value: 115
Dimensions: Server=Prod, Domain=Rio,       Unit: Count, Timestamp:
 2016-10-31T12:32:00Z, Value: 95
Dimensions: Server=Beta, Domain=Rio,       Unit: Count, Timestamp:
 2016-10-31T12:33:00Z, Value: 97
```

Wenn Sie nur die vier Metriken veröffentlichen, können Sie Statistiken für diese Kombinationen von Dimensionen abrufen:

- `Server=Prod,Domain=Frankfurt`
- `Server=Prod,Domain=Rio`
- `Server=Beta,Domain=Frankfurt`
- `Server=Beta,Domain=Rio`

Sie können dann keine Statistiken für die folgenden Dimensionen abrufen, wobei dasselbe gilt, wenn Sie keine Dimensionen angeben. (Die Ausnahme besteht darin, die metrische mathematische SEARCH-Funktion zu verwenden, die Statistiken für mehrere Metriken abrufen kann. Weitere Informationen finden Sie unter Suchausdrücke in Diagrammen verwenden.)

- `Server=Prod`
- `Server=Beta`
- `Domain=Frankfurt`
- `Domain=Rio`

Auflösung

Jede Metrik entspricht einer der folgenden:

- Standardauflösung; hierbei haben die Daten eine Granularität von einer Minute
- Hohe Auflösung; hierbei haben die Daten eine Granularität von einer Sekunde

Von AWS Diensten erzeugte Metriken haben standardmäßig die Standardauflösung. Wenn Sie eine benutzerdefinierte Metrik veröffentlichen, hat diese entweder die Standardauflösung oder eine hohe Auflösung. Wenn Sie eine Metrik mit hoher Auflösung veröffentlichen, wird sie mit einer Auflösung von 1 Sekunde CloudWatch gespeichert, sodass Sie sie mit einem Zeitraum von 1 Sekunde, 5 Sekunden, 10 Sekunden, 30 Sekunden oder einem beliebigen Vielfachen von 60 Sekunden lesen und abrufen können.

Mit hochauflösenden Metriken erhalten Sie genauere Einblicke in die Aktivitäten Ihrer Anwendung, die unter einer Minute liegen. Denken Sie daran, dass jeder `PutMetricData`-Aufruf einer benutzerdefinierten Metrik in Rechnung gestellt wird, sodass höhere Gebühren entstehen

können, wenn Sie häufiger `PutMetricData`-Aufrufe hochauflösender Metrik ausführen. Weitere Informationen zur CloudWatch Preisgestaltung finden Sie unter CloudWatch Amazon-Preise.

Wenn Sie einen Alarm für eine hochauflösende Metrik festlegen, können Sie einen hochauflösenden Alarm für einen Zeitraum von 10 Sekunden oder 30 Sekunden oder einen regelmäßigen Alarm für einen Zeitraum festlegen, der ein Mehrfaches von 60 Sekunden beträgt. Die Gebühr für hochauflösende Alarme mit einem Zeitraum von 10 oder 30 Sekunden ist höher.

Statistiken

Statistiken sind Aggregationen von metrischen Daten über bestimmte Zeiträume. CloudWatch stellt Statistiken bereit, die auf den metrischen Datenpunkten basieren, die durch Ihre benutzerdefinierten Daten oder durch andere AWS Dienste bereitgestellt werden. CloudWatch Für die Aggregationen werden der Namespace, der Metrikname, die Dimensionen und die Datenpunkt-Maßeinheit innerhalb des von Ihnen angegebenen Zeitraums verwendet.

Ausführliche Definitionen der Statistiken, die von unterstützt werden CloudWatch, finden Sie unterCloudWatch Definitionen von Statistiken.

Einheiten

Jede Statistik verfügt über eine Maßeinheit. Zu den Einheiten gehören beispielsweise `Bytes`, `Seconds`, `Count` und `Percent`. Eine vollständige Liste der CloudWatch unterstützten Einheiten finden Sie unter dem MetricDatumDatentyp in der Amazon CloudWatch API-Referenz.

Sie können beim Erstellen einer benutzerdefinierten Metrik eine Einheit angeben. Wenn Sie keine Einheit angeben, CloudWatch wird `None` als Einheit verwendet. Mit Einheiten können Sie Ihren Daten eine begriffliche Bedeutung verleihen. Obwohl einer Einheit CloudWatch intern keine Bedeutung beigemessen wird, können andere Anwendungen semantische Informationen auf der Grundlage der Einheit ableiten.

Metrik-Datenpunkte, die eine Maßeinheit angeben, werden separat aggregiert. Wenn Sie Statistiken abrufen, ohne eine Einheit anzugeben, werden CloudWatch alle Datenpunkte derselben Einheit zusammengefasst. Wenn Sie über zwei ansonsten identische Metriken mit unterschiedlichen Einheiten verfügen, werden zwei separate Datenstreams zurückgegeben, ein Datenstream für jede Einheit.

Zeiträume

Ein Zeitraum ist die Zeitdauer, die mit einer bestimmten CloudWatch Amazon-Statistik verknüpft ist. Jede Statistik stellt eine Aggregation der Metrikdaten dar, die über einen bestimmten Zeitraum erfasst wurden. Zeiträume sind in Anzahl Sekunden definiert, wobei es folgende gültige Werte für Zeiträume gibt: 1, 5, 10, 30 oder ein Vielfaches von 60. Um beispielsweise einen Zeitraum von sechs Minuten anzugeben, verwenden Sie den Zeitraumwert 360. Sie können anpassen, wie die Daten aggregiert werden, indem Sie die Länge des Zeitraums variieren. Der Standardwert eines Zeitraums ist 60 Sekunden. Ein Zeitraum kann nur eine Sekunde lang sein und muss ein Vielfaches von 60 sein, wenn er größer als der Standardwert von 60 Sekunden ist.

Nur benutzerdefinierte Metriken, die Sie mit einer Speicherauflösung von 1 Sekunde definieren unterstützen Zeiträume von weniger als einer Minute. Obwohl die Option, einen Zeitraum auf weniger als 60 einzustellen, immer verfügbar ist, sollten Sie in der Konsole einen Zeitraum wählen, welcher der Speicherung der Metrik entspricht. Weitere Informationen über Metriken, die Zeiträume von weniger als einer Minute unterstützen, finden Sie unter Hochauflösende Metriken.

Wenn Sie Statistiken abrufen, können Sie einen Zeitraum sowie eine Start- und Endzeit angeben. Diese Parameter bestimmen die allgemeine mit den Statistiken verbundene Dauer. Über die Standardwerte für die Startzeit und Endzeit erhalten Sie die Statistiken der letzten Stunde. Die Werte, die Sie für die Start- und Endzeit angeben, bestimmen, wie viele Perioden CloudWatch zurückgegeben werden. Das Abrufen von Statistiken unter Verwendung der Standardwerte für den Zeitraum, die Start- und Endzeit gibt beispielsweise eine aggregierte Reihe von Statistiken für jede Minute der vorherigen Stunde zurück. Wenn Sie lieber Statistiken haben möchten, die in Blöcke von 10 Minuten zusammengefasst sind, geben Sie einen Zeitraum von 600 an. Für Statistiken, die über die gesamte Stunde zusammengefasst sind, geben Sie einen Zeitraum von 3600 an.

Wenn Statistiken über einen bestimmten Zeitraum zusammengefasst werden, erhalten sie einen Zeitstempel mit dem Zeitpunkt des Beginns des Zeitraums. Beispiel: Daten, die zwischen 19:00 Uhr und 20:00 Uhr zusammengefasst werden, erhalten den Zeitstempel 19:00 Uhr. Darüber hinaus werden Daten, die zwischen 19:00 Uhr und 20:00 Uhr aggregiert wurden, ab 19:00 Uhr sichtbar. Dann können sich die Werte dieser aggregierten Daten ändern, wenn während des Zeitraums mehr Stichproben CloudWatch gesammelt werden.

Perioden sind auch wichtig für Alarme. CloudWatch Wenn Sie einen Alarm zur Überwachung einer bestimmten Metrik erstellen, bitten Sie CloudWatch darum, diese Metrik mit dem von Ihnen angegebenen Schwellenwert zu vergleichen. Sie haben umfassende Kontrolle darüber, wie CloudWatch dieser Vergleich durchgeführt wird. Sie können nicht nur den Zeitraum angeben, über

den der Vergleich erfolgen soll, sondern können auch angeben, wie viele Bewertungszeiträume verwendet werden sollen, um zu einer Schlussfolgerung zu gelangen. Wenn Sie beispielsweise drei Bewertungszeiträume angeben, wird ein Fenster mit drei Datenpunkten CloudWatch verglichen. CloudWatch benachrichtigt Sie nur, wenn bei dem ältesten Datenpunkt ein Verstoß vorliegt und bei den anderen Datenpunkten ein Verstoß vorliegt oder nicht vorhanden ist.

Aggregation

Amazon CloudWatch aggregiert Statistiken entsprechend der Periodenlänge, die Sie beim Abrufen der Statistiken angeben. Sie können beliebig viele Datenpunkte mit denselben oder ähnlichen Zeitstempeln veröffentlichen. CloudWatch aggregiert sie entsprechend der angegebenen Periodenlänge. CloudWatch aggregiert Daten nicht automatisch regionsübergreifend, aber Sie können Metrikmathematik verwenden, um Metriken aus verschiedenen Regionen zu aggregieren.

Sie können Datenpunkte für eine Metrik veröffentlichen, die nicht nur denselben Zeitstempel, sondern auch denselben Namespace und dieselben Dimensionen haben. CloudWatch gibt aggregierte Statistiken für diese Datenpunkte zurück. Sie können auch mehrere Datenpunkte für dieselben oder für verschiedene Metriken mit einem beliebigen Zeitstempel veröffentlichen.

Bei großen Datenmengen können Sie eine vorab aggregierten Datenmenge, die als Statistikgruppe bezeichnet wird, einfügen. Bei Statistiksätzen geben CloudWatch Sie Min., Max, Summe und SampleCount für eine Reihe von Datenpunkten an. Dies wird üblicherweise verwendet, wenn Sie mehrmals pro Minute Daten erfassen müssen. Angenommen, Sie haben beispielsweise eine Metrik für die Anfragelatenz einer Webseite. Es ist nicht sinnvoll, Daten bei jedem Webseitentreffer zu veröffentlichen. Wir empfehlen Ihnen, die Latenz aller Zugriffe auf diese Webseite zu erfassen, sie einmal pro Minute zu aggregieren und diesen Statistiksatz an zu senden. CloudWatch

Amazon unterscheidet CloudWatch nicht zwischen der Quelle einer Metrik. Wenn Sie eine Metrik mit demselben Namespace und denselben Dimensionen aus verschiedenen Quellen veröffentlichen, wird diese als eine einzige Metrik CloudWatch behandelt. Dies kann bei Service-Metriken in einem verteilten und skalierten System von Nutzen sein. Beispielsweise könnten alle Hosts in einer Webserver-Anwendung identische Metriken veröffentlichen, die die Latenz der Anfragen darstellen, die sie verarbeiten. CloudWatchbehandelt diese als eine einzige Metrik, sodass Sie die Statistiken für Minimum, Maximum, Durchschnitt und Summe aller Anfragen in Ihrer Anwendung abrufen können.

Perzentile

Ein Perzentil gibt die relative Stelle eines Wertes in einer Datenmenge an. Das 95. Perzentil bedeutet beispielsweise, dass 95 Prozent der Daten unter diesem Wert liegen und 5 Prozent der Daten über

diesem Wert. Perzentile verhelfen Ihnen zu einem besseren Verständnis für die Verteilung Ihrer Metrikdaten.

Perzentile werden häufig genutzt, um Anomalien zu isolieren. In einer normalen Verteilung befinden sich 95 Prozent der Daten innerhalb von zwei Standardabweichungen vom Mittelwert und 99,7 Prozent der Daten innerhalb von drei Standardabweichungen vom Mittelwert. Alle Daten, die außerhalb von drei Standardabweichungen liegen, werden häufig als Anomalie betrachtet, da sie so enorm vom Durchschnittswert abweichen. Angenommen, Sie überwachen beispielsweise die CPU-Auslastung Ihrer EC2-Instances, um sicherzustellen, dass Ihre Kunden eine gute Benutzererfahrung machen. Wenn Sie nur den Durchschnittswert überwachen, können Anomalien verborgen bleiben. Wenn Sie den Maximalwert überwachen, kann eine einzelne Anomalie die Ergebnisse verzerren. Mit Perzentilen können Sie das 95. Perzentil der CPU-Auslastung überwachen, um Instances auf eine ungewöhnlich hohe Belastung hin zu überprüfen.

Einige CloudWatch Metriken unterstützen Perzentile als Statistik. Für diese Metriken können Sie Ihr System und Ihre Anwendungen mithilfe von Perzentilen überwachen, genauso wie Sie es mit den anderen CloudWatch Statistiken (Durchschnitt, Minimum, Maximum und Summe) tun würden. Sie können beispielsweise beim Erstellen eines Alarms Perzentile als statistische Funktion verwenden. Sie können die Perzentile mit bis zu zehn Dezimalstellen angeben (z. B. p95,0123456789).

Perzentil-Statistiken sind für benutzerdefinierte Metriken verfügbar, solange Sie die unformatierten, nicht zusammengefassten Datenpunkte für Ihre benutzerdefinierte Metrik veröffentlichen. Perzentil-Statistiken für Metriken sind nicht verfügbar, wenn es Metrik-Werte gibt, die negative Zahlen enthalten.

CloudWatch benötigt Rohdatenpunkte, um Perzentile zu berechnen. Wenn Sie Daten stattdessen mit einer Statistikgruppe veröffentlichen, können Sie nur dann eine Perzentil-Statistik für diese Daten abrufen, wenn eine der folgenden Bedingungen erfüllt ist:

- Der SampleCount Wert des Statistiksatzes ist 1 und Min, Max und Summe sind alle gleich.
- Min und Max sind gleich, und Summe ist gleich Min multipliziert mit. SampleCount

Die folgenden AWS Dienste umfassen Metriken, die Perzentilstatistiken unterstützen.

- API Gateway
- Application Load Balancer
- Amazon EC2
- Elastic Load Balancing

- Kinesis
- Amazon RDS

CloudWatch unterstützt auch Statistiken zum getrimmten Mittelwert und andere Leistungsstatistiken, die ähnlich wie Perzentile verwendet werden können. Weitere Informationen finden Sie unter CloudWatch Definitionen von Statistiken.

Alarme

Sie können einen Alarm verwenden, um Aktionen in Ihrem Namen automatisch zu initiieren. Ein Alarm überwacht eine Metrik über einen bestimmten Zeitraum und führt eine oder mehrere festgelegte Aktionen durch, die auf dem Wert der Metrik im Verhältnis zu einem bestimmten zeitlichen Schwellenwert basieren. Die Aktion ist eine Benachrichtigung, die an ein Amazon-SNS-Thema oder eine Auto-Scaling-Richtlinie gesendet wird. Sie können auch Alarme zu Dashboards hinzufügen.

Bei Alarmen werden nur Aktionen für anhaltende Statusänderungen ausgelöst. CloudWatch Alarme lösen keine Aktionen aus, nur weil sie sich in einem bestimmten Zustand befinden. Der Status muss sich geändert haben und für eine festgelegte Anzahl an Zeiträumen aufrechterhalten worden sein.

Wählen Sie beim Erstellen eines Alarms einen Zeitraum für die Alarmüberwachung aus, der größer als der oder gleich der Auflösung der Metrik ist. Die grundlegende Überwachung für Amazon EC2 stellt beispielsweise alle 5 Minuten Metriken für Ihre Instances zur Verfügung. Wählen Sie beim Einstellen eines Alarms für eine grundlegende Überwachungsmetrik einen Zeitraum von mindestens 300 Sekunden (5 Minuten). Detaillierte Überwachung für Amazon EC2 stellt Metriken für Ihre Instances mit einer Auflösung von 1 Minute zur Verfügung. Wählen Sie beim Einstellen eines Alarms für eine detaillierte Überwachungsmetrik einen Zeitraum von mindestens 60 Sekunden (1 Minute).

Wenn Sie einen Alarm für eine hochauflösende Metrik festlegen, können Sie einen hochauflösenden Alarm für einen Zeitraum von 10 Sekunden oder 30 Sekunden oder einen regelmäßigen Alarm für einen Zeitraum festlegen, der ein Mehrfaches von 60 Sekunden beträgt. Für hochauflösende Alarme ist eine höhere Gebühr zu zahlen. Weitere Informationen zu hochauflösenden Metriken finden Sie unter Veröffentlichen von benutzerdefinierten -Metriken.

Weitere Informationen finden Sie unter CloudWatch Amazon-Alarme verwenden und Einen Alarm aus einer Metrik in einem Diagramm erstellen.

Fakturierung und Kosten

Vollständige Informationen zu den CloudWatch Preisen finden Sie unter CloudWatch Amazon-Preise.

Informationen, die Ihnen helfen können, Ihre Rechnung zu analysieren und möglicherweise Kosten zu optimieren und zu senken, finden Sie unter CloudWatch Abrechnung und Kosten.

CloudWatch Amazon-Ressourcen

Die folgenden verwandten Ressourcen bieten Ihnen nützliche Informationen für die Arbeit mit diesem Service.

Ressource	Beschreibung
CloudWatchHäufig gestellte Fragen zu Amazon	Die Webseite „Häufig gestellte Fragen” deckt alle wichtigsten Fragen ab, die Entwickler zu diesem Produkt gestellt haben.
AWS Entwicklerzentrum	Ein zentraler Ausgangspunkt, um Dokumentation, Codebeispiele, Versionshinweise und andere Informati onen zu finden, mit denen Sie innovative Anwendungen entwickeln können AWS.
AWS Management Console	Mit der Konsole können Sie die meisten Funktionen von Amazon CloudWatch und verschiedenen anderen AWS Angeboten ohne Programmierung ausführen.
CloudWatch Amazon-Diskussions foren	Community-basiertes Forum für Entwickler zur Erörterung technischer Fragen zu Amazon. CloudWatch
AWS Support	Die zentrale Anlaufstelle für die Erstellung und Verwaltun g Ihrer AWS Support Fälle. Enthält auch Links zu anderen hilfreichen Ressourcen wie Foren, häufig gestellten technischen Fragen, dem Status des Dienstes und AWS Trusted Advisor.

Ressource	Beschreibung
CloudWatch Amazon-Produktinformationen	Die primäre Webseite für Informationen über Amazon CloudWatch.
Kontakt	Eine zentrale Anlaufstelle für Anfragen zu AWS Rechnungen, Konten, Veranstaltungen, Missbrauch usw.

Einrichten

Um Amazon nutzen zu CloudWatch können, benötigen Sie ein AWS Konto. Mit Ihrem AWS Konto können Sie Dienste (z. B. Amazon EC2) verwenden, um Metriken zu generieren, die Sie in der CloudWatch Konsole, einer point-and-click webbasierten Oberfläche, anzeigen können. Darüber hinaus können Sie die AWS Befehlszeilenschnittstelle (CLI) installieren und konfigurieren.

Melden Sie sich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/signup.
2. Folgen Sie den Online-Anweisungen.

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Aus Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich ist.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu https://aws.amazon.com/ auf und klicken Sie auf Mein Konto.

Erstellen Sie einen Benutzer mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS Management Consoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen Sie einen Benutzer mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.
2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis im AWS IAM Identity Center Benutzerhandbuch.

Melden Sie sich als Benutzer mit Administratorzugriff an

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Zugriffsportal.

Weisen Sie weiteren Benutzern Zugriff zu

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

Anweisungen finden Sie im Benutzerhandbuch unter Einen Berechtigungssatz erstellen.AWS IAM Identity Center

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Anweisungen finden Sie im AWS IAM Identity Center Benutzerhandbuch unter Gruppen hinzufügen.

Melden Sie sich bei der CloudWatch Amazon-Konsole an

So melden Sie sich bei der CloudWatch Amazon-Konsole an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Verwenden Sie bei Bedarf die Navigationsleiste, um die Region auf die Region umzustellen, in der Sie Ihre AWS Ressourcen haben.
3. Selbst wenn Sie die CloudWatch Konsole zum ersten Mal verwenden, könnte Your Metrics bereits Kennzahlen melden, da Sie ein AWS Produkt verwendet haben, das Metriken automatisch und CloudWatch kostenlos an Amazon überträgt. Bei anderen Services müssen Sie Metriken aktivieren.

   Wenn Sie keine Alarme erstellt haben, enthält der Abschnitt Your Alarms die Schaltfläche Create Alarm.

Richten Sie das ein AWS CLI

Sie können die AWS CLI oder die CloudWatch Amazon-CLI verwenden, um CloudWatch Befehle auszuführen. Beachten Sie, dass das die CloudWatch CLI AWS CLI ersetzt; wir nehmen neue CloudWatch Funktionen nur in die auf AWS CLI.

Informationen zur Installation und Konfiguration von finden Sie unter Getting Up with the AWS Command Line Interface im AWS Command Line Interface Benutzerhandbuch. AWS CLI

Informationen zur Installation und Konfiguration der Amazon CloudWatch CLI finden Sie unter Setup the Command Line Interface in der Amazon CloudWatch CLI Reference.

Erste Schritte mit Amazon CloudWatch

Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

Die CloudWatch Übersichts-Startseite wird angezeigt.

Die Übersicht zeigt die folgenden Elemente, die automatisch aktualisiert werden.

- Alarme nach AWS Dienst zeigt eine Liste der AWS Dienste an, die Sie in Ihrem Konto verwenden, zusammen mit dem Status der Alarme in diesen Diensten. Daneben werden zwei oder vier Alarme in Ihrem Konto angezeigt. Die Anzahl hängt davon ab, wie viele AWS Dienste Sie nutzen. Die angezeigten Alarme sind diejenigen im ALARM-Zustand oder diejenigen, die den Zustand zuletzt geändert haben.

  Diese oberen Bereiche helfen Ihnen dabei, den Zustand Ihrer AWS Dienste schnell zu beurteilen, da sie die Alarmstatus in jedem Dienst und die Alarme, deren Status zuletzt geändert wurde, einsehen können. Auf diese Weise können Sie Probleme überwachen und schnell diagnostizieren.
- Unter diesen Bereichen befindet sich das Standard-Dashboard, sofern vorhanden. Das Standard-Dashboard ist ein benutzerdefiniertes Dashboard, das Sie erstellt und CloudWatch-Default genannt haben. Auf diese Weise können Sie der Übersichtsseite bequem Metriken zu Ihren eigenen benutzerdefinierten Diensten oder Anwendungen hinzufügen oder zusätzliche wichtige Metriken von AWS Diensten anzeigen, die Sie am meisten überwachen möchten.

> **Note**
>
> In den automatischen Dashboards auf der CloudWatch Startseite werden nur Informationen aus dem Girokonto angezeigt, auch wenn es sich bei dem Konto um ein Überwachungskonto handelt, das für CloudWatch kontoübergreifende Beobachtbarkeit eingerichtet wurde. Informationen zum Erstellen kontenübergreifender Dashboards finden Sie unter CloudWatch Dashboard zur kontenübergreifenden Beobachtbarkeit.

In dieser Übersicht können Sie sich ein dienstübergreifendes Dashboard mit Kennzahlen verschiedener AWS Services anzeigen lassen oder sich auf eine bestimmte Ressourcengruppe oder einen bestimmten Service konzentrieren. AWS Auf diese Weise können Sie Ihre Ansicht auf eine Teilmenge von Ressourcen einschränken, an denen Sie interessiert sind. Weitere Informationen finden Sie in den folgenden Abschnitten.

Sehen Sie sich das automatische vorgefertigte Dashboard für einen einzelnen Service an

Um das automatische vorgefertigte Dashboard für einen einzelnen Service zu sehen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

   Die Startseite wird angezeigt.
2. Wählen Sie im linken Navigationsbereich Dashboards aus.
3. Wählen Sie die Registerkarte Automatische Dashboards und dann den Dienst aus, den Sie sehen möchten.
4. Um zur Anzeige der Alarme für diesen Dienst zu wechseln, aktivieren Sie oben auf dem Bildschirm, wo der Dienstname derzeit angezeigt wird, das Kontrollkästchen für Bei Alarm, Ungenügende Daten oder OK.
5. Wenn Sie Metriken anzeigen, können Sie sich auf verschiedene Arten auf eine bestimmte Metrik konzentrieren:

   a. Um weitere Details zu den Metriken in einem beliebigen Diagramm anzuzeigen, fahren Sie mit der Maus über das Diagramm und wählen Sie das Aktionssymbol aus, View in metrics (In Metriken anzeigen).

Das Diagramm erscheint in einer neuen Registerkarte, wobei die relevanten Metriken unterhalb des Diagramms aufgelistet sind. Sie können Ihre Ansicht dieses Diagramms anpassen, indem Sie die angezeigten Metriken und Ressourcen, die Statistik, den Zeitraum und andere Faktoren ändern, um ein besseres Verständnis der aktuellen Situation zu erhalten.

b. Sie können Protokollereignisse aus dem im Diagramm angezeigten Zeitraum anzeigen. Dies kann Ihnen helfen, Ereignisse in Ihrer Infrastruktur zu erkennen, die zu einer unerwarteten Änderung Ihrer Metriken führen.

   Um die Protokollereignisse anzuzeigen, fahren Sie mit der Maus über das Diagramm und wählen Sie das Aktionssymbol aus, View in logs (In Protokollen anzeigen).

   Die Ansicht „ CloudWatch Protokolle“ wird auf einer neuen Registerkarte mit einer Liste Ihrer Protokollgruppen angezeigt. Um die Protokollereignisse in einer dieser Protokollgruppen anzuzeigen, die während des im Originaldiagramm angezeigten Zeitraums aufgetreten sind, wählen Sie diese Protokollgruppe aus.

6. Wenn Sie Alarme anzeigen, können Sie sich auf verschiedene Arten auf einen bestimmten Alarm konzentrieren:

   • Um weitere Details zu einem Alarm zu sehen, fahren Sie mit der Maus über den Alarm und wählen Sie das Aktionssymbol aus, View in alarms (In Alarmen anzeigen).

     Die Alarmansicht erscheint in einer neuen Registerkarte und zeigt eine Liste Ihrer Alarme sowie Details zum ausgewählten Alarm an. Um den Verlauf für diesen Alarm anzuzeigen, wählen Sie die Registerkarte History (Verlauf) aus.

7. Alarme werden immer einmal pro Minute aktualisiert. Um die Ansicht zu aktualisieren, wählen Sie das Aktualisierungssymbol (zwei gebogene Pfeile) oben rechts auf dem Bildschirm aus. Um die automatische Aktualisierungsrate für andere Elemente auf dem Bildschirm als Alarme zu ändern, wählen Sie den Pfeil nach unten neben dem Aktualisierungssymbol aus und klicken Sie auf eine Aktualisierungsrate. Sie können die automatische Aktualisierung auch deaktivieren.
8. Um den Zeitraum zu ändern, der in allen aktuell angezeigten Diagrammen und Alarmen angezeigt wird, wählen Sie oben auf dem Bildschirm, neben Time range (Zeitraum) den gewünschten Zeitraum aus. Um aus mehr Zeitraumsoptionen auszuwählen, als standardmäßig angezeigt werden, wählen Sie custom (benutzerdefiniert) aus.

9. Um zum serviceübergreifenden Dashboard zurückzukehren, wählen Sie Overview (Übersicht) in der Liste oben auf dem Bildschirm aus, die derzeit den Service anzeigt, auf den Sie sich konzentrieren.

   Alternativ können Sie in einer beliebigen Ansicht oben CloudWatchauf dem Bildschirm auswählen, ob Sie alle Filter löschen und zur Übersichtsseite zurückkehren möchten.

Sehen Sie sich das vorgefertigte dienstübergreifende Dashboard an

Sie können zum serviceübergreifenden Dashboard-Bildschirm wechseln und mit den Dashboards für alle AWS Dienste interagieren, die Sie verwenden. In der CloudWatch Konsole werden Ihre Dashboards in alphabetischer Reihenfolge angezeigt und auf jedem Dashboard werden ein oder zwei wichtige Kennzahlen angezeigt.

> **Note**
>
> Wenn Sie fünf oder mehr AWS Dienste verwenden, zeigt die CloudWatch Konsole das dienstübergreifende Dashboard nicht auf dem Übersichtsbildschirm an.

So öffnen Sie das serviceübergreifende Dashboard:

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

   Sie werden zum Übersichtsbildschirm weitergeleitet.
2. Wählen Sie auf dem Übersichtsbildschirm das Dropdown-Menü Overview (Übersicht) und dann Cross service dashboard (Serviceübergreifendes Dashboard) aus.

   Sie werden zum Bildschirm „Serviceübergreifendes Dashboard“ weitergeleitet.
3. (Optional) Wenn Sie die ursprüngliche Schnittstelle verwenden, scrollen Sie zum Abschnitt Cross-service dashboard (Serviceübergreifendes Dashboard) und wählen Sie dann View Cross-service dashboard (Serviceübergreifendes Dashboard anzeigen) aus.

   Sie werden zum Bildschirm „Serviceübergreifendes Dashboard“ weitergeleitet.
4. Sie können sich auf zwei Arten auf einen bestimmten Service konzentrieren:

   a. Um weitere Schlüsselmetriken für einen Service anzuzeigen, wählen Sie seinen Namen aus der Liste am oberen Bildschirmrand, wo derzeit Cross Service Dashboard

(Serviceübergreifendes Dashboard) angezeigt wird. Alternativ können Sie neben dem Namen des Services auf View Service dashboard (Service-Dashboard anzeigen) klicken.

Es wird ein automatisches Dashboard für diesen Service angezeigt, das weitere Metriken für diesen Service anzeigt. Darüber hinaus werden für einige Services am unteren Rand des Service-Dashboards Ressourcen angezeigt, die sich auf diesen Service beziehen. Sie können eine dieser Ressourcen für diese Servicekonsole auswählen und sich weiter auf diese Ressource konzentrieren.

b. Um alle Alarme im Zusammenhang mit einem Service anzuzeigen, wählen Sie die Schaltfläche auf der rechten Seite des Bildschirms neben diesem Servicenamen aus. Der Text auf dieser Schaltfläche gibt an, wie viele Alarme Sie in diesem Service erstellt haben und ob sich diese im ALARM-Zustand befinden.

Wenn die Alarme angezeigt werden, können mehrere Alarme mit ähnlichen Einstellungen (wie Dimensionen, Schwellenwert oder Zeitraum) in einem einzigen Diagramm angezeigt werden.

Sie können dann Details zu einem Alarm anzeigen und den Alarmverlauf einsehen. Bewegen Sie dazu den Mauszeiger über das Alarm-Diagramm und wählen Sie das Aktionssymbol, View in alarms (In Alarmen anzeigen) aus.

Die Alarmansicht erscheint in einer neuen Registerkarte des Browsers und zeigt eine Liste Ihrer Alarme sowie Details zum ausgewählten Alarm an. Um den Verlauf für diesen Alarm anzuzeigen, wählen Sie die Registerkarte History (Verlauf) aus.

5. Sie können sich auf Ressourcen in einer bestimmten Ressourcengruppe konzentrieren. Wählen Sie dazu die Ressourcengruppe aus der Liste oben auf der Seite aus, wo All resources (Alle Ressourcen) angezeigt wird.

   Weitere Informationen finden Sie unter Sehen Sie sich ein vorgefertigtes Dashboard für eine Ressourcengruppe an.

6. Um den Zeitraum zu ändern, der in allen aktuell angezeigten Diagrammen und Alarmen angezeigt wird, wählen Sie den gewünschten Bereich neben Time range (Zeitraum) oben auf dem Bildschirm aus. Wählen Sie custom (benutzerdefiniert) aus, um aus mehr Zeitraumoptionen auszuwählen, als standardmäßig angezeigt werden.

7. Alarme werden immer einmal pro Minute aktualisiert. Um die Ansicht zu aktualisieren, wählen Sie das Aktualisierungssymbol (zwei gebogene Pfeile) oben rechts auf dem Bildschirm aus. Um die automatische Aktualisierungsrate für andere Elemente auf dem Bildschirm als Alarme zu ändern,

wählen Sie den Pfeil nach unten neben dem Aktualisierungssymbol aus und klicken Sie auf die gewünschte Aktualisierungsrate. Sie können die automatische Aktualisierung auch deaktivieren.

Entfernen Sie einen Dienst aus dem dienstübergreifenden Dashboard

Sie können verhindern, dass die Metriken eines Services im serviceübergreifenden Dashboard angezeigt werden. Auf diese Weise können Sie sich im serviceübergreifenden Dashboard auf die Services konzentrieren, die Sie am häufigsten überwachen möchten.

Wenn Sie einen Service aus dem serviceübergreifenden Dashboard entfernen, werden die Alarme für diesen Service weiterhin in den Ansichten Ihrer Alarme angezeigt.

So entfernen Sie die Metriken eines Services aus dem serviceübergreifenden Dashboard

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

   Die Startseite wird angezeigt.
2. Wählen Sie am oberen Rand der Seite, unter Overview (Übersicht), den Service aus, den Sie entfernen möchten.

   Die Ansicht ändert sich, um Metriken nur von diesem Service anzuzeigen.
3. Wählen Sie Actions (Aktionen) aus, und deaktivieren Sie dann das Kontrollkästchen neben Show on cross service dashboard (Auf dem serviceübergreifenden Dashboard anzeigen).

Sehen Sie sich ein vorgefertigtes Dashboard für eine Ressourcengruppe an

Sie können Ihre Ansicht konzentrieren, um Metriken und Alarme aus einer einzigen Ressourcengruppe anzuzeigen. Mit Ressourcengruppen können Sie mit Hilfe von Tags Projekte organisieren, sich auf eine Teilmenge Ihrer Architektur konzentrieren oder zwischen Ihren Produktions- und Entwicklungsumgebungen unterscheiden. Sie ermöglichen es Ihnen auch, sich in der CloudWatch Übersicht auf jede dieser Ressourcengruppen zu konzentrieren. Weitere Informationen finden Sie unter Was ist AWS Resource Groups?.

Wenn Sie sich auf eine Ressourcengruppe konzentrieren, ändert sich die Anzeige, um nur die Services anzuzeigen, bei denen Sie als Teil dieser Ressourcengruppe Ressourcen markiert haben. Der Bereich für die letzten Alarme zeigt nur Alarme an, die Ressourcen zugeordnet sind, die Teil der Ressourcengruppe sind. Wenn Sie ein Dashboard mit dem Namen CloudWatch-Default- erstellt haben ResourceGroupName, wird es außerdem im Standard-Dashboard-Bereich angezeigt.

Sie können weiter nach unten gehen, indem Sie sich gleichzeitig auf einen einzelnen AWS Service und eine Ressourcengruppe konzentrieren. Das folgende Verfahren erklärt nur, wie Sie sich auf eine Ressourcengruppe konzentrieren können.

So konzentrieren Sie sich auf eine einzelne Ressourcengruppe

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Oben auf der Seite, wo die All resources (Alle Ressourcen) angezeigt wird, wählen Sie eine Ressourcengruppe aus.
3. Um weitere Metriken zu dieser Ressourcengruppe am unteren Bildschirmrand anzuzeigen, wählen Sie View cross service dashboard (Serviceübergreifendes Dashboard anzeigen) aus.

   Das serviceübergreifende Dashboard wird angezeigt und führt nur die Services auf, die sich auf diese Ressourcengruppe beziehen. Für jeden Service werden ein oder zwei Schlüsselmetriken angezeigt.
4. Um den Zeitraum zu ändern, der in allen aktuell angezeigten Diagrammen und Alarmen angezeigt wird, wählen Sie neben Time range (Zeitraum) oben auf dem Bildschirm den gewünschten Zeitraum aus. Um aus mehr Zeitraumsoptionen auszuwählen, als standardmäßig angezeigt werden, wählen Sie custom (benutzerdefiniert) aus.
5. Alarme werden immer einmal pro Minute aktualisiert. Um die Ansicht zu aktualisieren, wählen Sie das Aktualisierungssymbol (zwei gebogene Pfeile) oben rechts auf dem Bildschirm aus. Um die automatische Aktualisierungsrate für andere Elemente auf dem Bildschirm als Alarme zu ändern, wählen Sie den Pfeil nach unten neben dem Aktualisierungssymbol aus und klicken Sie auf eine Aktualisierungsrate. Sie können die automatische Aktualisierung auch deaktivieren.
6. Um zur Anzeige von Informationen über alle Ressourcen in Ihrem Konto zurückzukehren, wählen Sie oben auf dem Bildschirm, wo aktuell der Name der Ressourcengruppe angezeigt wird, All resources (Alle Ressourcen) aus.

Sehen Sie sich das vorgefertigte dienstübergreifende Dashboard an

Sie können zum dienstübergreifenden Dashboard-Bildschirm wechseln und mit den Dashboards für alle AWS Dienste interagieren, die Sie verwenden. Die CloudWatch Konsole zeigt Ihre Dashboards in alphabetischer Reihenfolge an und zeigt ein oder zwei wichtige Kennzahlen für jeden Service an.

> **Note**
>
> Wenn Sie fünf oder mehr AWS Dienste verwenden, zeigt die CloudWatch Konsole das dienstübergreifende Dashboard nicht auf dem Übersichtsbildschirm an.

So öffnen Sie das serviceübergreifende Dashboard:

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

   Sie werden zum Übersichtsbildschirm weitergeleitet.
2. Wählen Sie auf dem Übersichtsbildschirm das Dropdown-Menü Overview (Übersicht) und dann Cross service dashboard (Serviceübergreifendes Dashboard) aus.

   Sie werden zum Bildschirm „Serviceübergreifendes Dashboard“ weitergeleitet.
3. (Optional) Wenn Sie die ursprüngliche Schnittstelle verwenden, scrollen Sie zum Abschnitt Cross-service dashboard (Serviceübergreifendes Dashboard) und wählen Sie dann View Cross-service dashboard (Serviceübergreifendes Dashboard anzeigen) aus.

   Sie werden zum Bildschirm „Serviceübergreifendes Dashboard“ weitergeleitet.
4. Sie können sich auf zwei Arten auf einen bestimmten Service konzentrieren:

   a. Um weitere Schlüsselmetriken für einen Service anzuzeigen, wählen Sie seinen Namen aus der Liste am oberen Bildschirmrand, wo derzeit Cross Service Dashboard (Serviceübergreifendes Dashboard) angezeigt wird. Alternativ können Sie neben dem Namen des Services auf View Service dashboard (Service-Dashboard anzeigen) klicken.

      Es wird ein automatisches Dashboard für diesen Service angezeigt, das weitere Metriken für diesen Service anzeigt. Darüber hinaus werden für einige Services am unteren Rand des Service-Dashboards Ressourcen angezeigt, die sich auf diesen Service beziehen. Sie können eine dieser Ressourcen für diese Servicekonsole auswählen und sich weiter auf diese Ressource konzentrieren.

   b. Um alle Alarme im Zusammenhang mit einem Service anzuzeigen, wählen Sie die Schaltfläche auf der rechten Seite des Bildschirms neben diesem Servicenamen aus. Der Text auf dieser Schaltfläche gibt an, wie viele Alarme Sie in diesem Service erstellt haben und ob sich diese im ALARM-Zustand befinden.

Wenn die Alarme angezeigt werden, können mehrere Alarme mit ähnlichen Einstellungen (wie Dimensionen, Schwellenwert oder Zeitraum) in einem einzigen Diagramm angezeigt werden.

Sie können dann Details zu einem Alarm anzeigen und den Alarmverlauf einsehen. Bewegen Sie dazu den Mauszeiger über das Alarm-Diagramm und wählen Sie das Aktionssymbol, View in alarms (In Alarmen anzeigen) aus.

Die Alarmansicht erscheint in einer neuen Registerkarte des Browsers und zeigt eine Liste Ihrer Alarme sowie Details zum ausgewählten Alarm an. Um den Verlauf für diesen Alarm anzuzeigen, wählen Sie die Registerkarte History (Verlauf) aus.

5. Sie können sich auf Ressourcen in einer bestimmten Ressourcengruppe konzentrieren. Wählen Sie dazu die Ressourcengruppe aus der Liste oben auf der Seite aus, wo All resources (Alle Ressourcen) angezeigt wird.

   Weitere Informationen finden Sie unter Sehen Sie sich ein vorgefertigtes Dashboard für eine Ressourcengruppe an.
6. Um den Zeitraum zu ändern, der in allen aktuell angezeigten Diagrammen und Alarmen angezeigt wird, wählen Sie den gewünschten Bereich neben Time range (Zeitraum) oben auf dem Bildschirm aus. Wählen Sie custom (benutzerdefiniert) aus, um aus mehr Zeitraumoptionen auszuwählen, als standardmäßig angezeigt werden.
7. Alarme werden immer einmal pro Minute aktualisiert. Um die Ansicht zu aktualisieren, wählen Sie das Aktualisierungssymbol (zwei gebogene Pfeile) oben rechts auf dem Bildschirm aus. Um die automatische Aktualisierungsrate für andere Elemente auf dem Bildschirm als Alarme zu ändern, wählen Sie den Pfeil nach unten neben dem Aktualisierungssymbol aus und klicken Sie auf die gewünschte Aktualisierungsrate. Sie können die automatische Aktualisierung auch deaktivieren.

Entfernen eines Services aus der Anzeige im serviceübergreifenden Dashboard

Sie können verhindern, dass die Metriken eines Services im serviceübergreifenden Dashboard angezeigt werden. Auf diese Weise können Sie sich im serviceübergreifenden Dashboard auf die Services konzentrieren, die Sie am häufigsten überwachen möchten.

Wenn Sie einen Service aus dem serviceübergreifenden Dashboard entfernen, werden die Alarme für diesen Service weiterhin in den Ansichten Ihrer Alarme angezeigt.

So entfernen Sie die Metriken eines Services aus dem serviceübergreifenden Dashboard

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

   Die Startseite wird angezeigt.
2. Wählen Sie am oberen Rand der Seite, unter Overview (Übersicht), den Service aus, den Sie entfernen möchten.

   Die Ansicht ändert sich, um Metriken nur von diesem Service anzuzeigen.
3. Wählen Sie Actions (Aktionen) aus, und deaktivieren Sie dann das Kontrollkästchen neben Show on cross service dashboard (Auf dem serviceübergreifenden Dashboard anzeigen).

Sehen Sie sich ein vorgefertigtes Dashboard für einen einzelnen AWS Service an

Auf der CloudWatch Startseite können Sie die Ansicht auf einen einzelnen AWS Dienst konzentrieren. Sie können weiter nach unten gehen, indem Sie sich gleichzeitig auf einen einzelnen AWS Service und eine Ressourcengruppe konzentrieren. Das folgende Verfahren zeigt nur, wie Sie sich auf einen AWS Service konzentrieren können.

So konzentrieren Sie sich auf einen einzelnen Service

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

   Die Startseite wird angezeigt.
2. Wählen Sie unter Übersicht, wo Übersicht derzeit im Dropdownmenü angezeigt wird, die Option Service-Dashboards aus.
3. Wählen Sie den Service aus, auf den Sie sich konzentrieren möchten.

   Die Ansicht ändert sich, um Diagramme von Schlüsselmetriken aus dem ausgewählten Service anzuzeigen.
4. Um zur Anzeige der Alarme für diesen Dienst zu wechseln, aktivieren Sie oben auf dem Bildschirm, wo der Dienstname derzeit angezeigt wird, das Kontrollkästchen für Bei Alarm, Ungenügende Daten oder OK.
5. Wenn Sie Metriken anzeigen, können Sie sich auf verschiedene Arten auf eine bestimmte Metrik konzentrieren:

a. Um weitere Details zu den Metriken in einem beliebigen Diagramm anzuzeigen, fahren Sie mit der Maus über das Diagramm und wählen Sie das Aktionssymbol aus, View in metrics (In Metriken anzeigen).

   Das Diagramm erscheint in einer neuen Registerkarte, wobei die relevanten Metriken unterhalb des Diagramms aufgelistet sind. Sie können Ihre Ansicht dieses Diagramms anpassen, indem Sie die angezeigten Metriken und Ressourcen, die Statistik, den Zeitraum und andere Faktoren ändern, um ein besseres Verständnis der aktuellen Situation zu erhalten.

b. Sie können Protokollereignisse aus dem im Diagramm angezeigten Zeitraum anzeigen. Dies kann Ihnen helfen, Ereignisse in Ihrer Infrastruktur zu erkennen, die zu einer unerwarteten Änderung Ihrer Metriken führen.

   Um die Protokollereignisse anzuzeigen, fahren Sie mit der Maus über das Diagramm und wählen Sie das Aktionssymbol aus, View in logs (In Protokollen anzeigen).

   Die Ansicht „ CloudWatch Protokolle“ wird auf einer neuen Registerkarte mit einer Liste Ihrer Protokollgruppen angezeigt. Um die Protokollereignisse in einer dieser Protokollgruppen anzuzeigen, die während des im Originaldiagramm angezeigten Zeitraums aufgetreten sind, wählen Sie diese Protokollgruppe aus.

6. Wenn Sie Alarme anzeigen, können Sie sich auf verschiedene Arten auf einen bestimmten Alarm konzentrieren:

   • Um weitere Details zu einem Alarm zu sehen, fahren Sie mit der Maus über den Alarm und wählen Sie das Aktionssymbol aus, View in alarms (In Alarmen anzeigen).

     Die Alarmansicht erscheint in einer neuen Registerkarte und zeigt eine Liste Ihrer Alarme sowie Details zum ausgewählten Alarm an. Um den Verlauf für diesen Alarm anzuzeigen, wählen Sie die Registerkarte History (Verlauf) aus.

7. Alarme werden immer einmal pro Minute aktualisiert. Um die Ansicht zu aktualisieren, wählen Sie das Aktualisierungssymbol (zwei gebogene Pfeile) oben rechts auf dem Bildschirm aus. Um die automatische Aktualisierungsrate für andere Elemente auf dem Bildschirm als Alarme zu ändern, wählen Sie den Pfeil nach unten neben dem Aktualisierungssymbol aus und klicken Sie auf eine Aktualisierungsrate. Sie können die automatische Aktualisierung auch deaktivieren.
8. Um den Zeitraum zu ändern, der in allen aktuell angezeigten Diagrammen und Alarmen angezeigt wird, wählen Sie oben auf dem Bildschirm, neben Time range (Zeitraum) den

gewünschten Zeitraum aus. Um aus mehr Zeitraumsoptionen auszuwählen, als standardmäßig angezeigt werden, wählen Sie custom (benutzerdefiniert) aus.

9. Um zum serviceübergreifenden Dashboard zurückzukehren, wählen Sie Overview (Übersicht) in der Liste oben auf dem Bildschirm aus, die derzeit den Service anzeigt, auf den Sie sich konzentrieren.

   Alternativ können Sie in einer beliebigen Ansicht oben CloudWatchauf dem Bildschirm auswählen, ob Sie alle Filter löschen und zur Übersichtsseite zurückkehren möchten.

Sehen Sie sich ein vorgefertigtes Dashboard für eine Ressourcengruppe an

Sie können Ihre Ansicht konzentrieren, um Metriken und Alarme aus einer einzigen Ressourcengruppe anzuzeigen. Mit Ressourcengruppen können Sie mit Hilfe von Tags Projekte organisieren, sich auf eine Teilmenge Ihrer Architektur konzentrieren oder zwischen Ihren Produktions- und Entwicklungsumgebungen unterscheiden. Sie ermöglichen es Ihnen auch, sich in der CloudWatch Übersicht auf jede dieser Ressourcengruppen zu konzentrieren. Weitere Informationen finden Sie unter Was ist AWS Resource Groups?.

Wenn Sie sich auf eine Ressourcengruppe konzentrieren, ändert sich die Anzeige, um nur die Services anzuzeigen, bei denen Sie als Teil dieser Ressourcengruppe Ressourcen markiert haben. Der Bereich für die letzten Alarme zeigt nur Alarme an, die Ressourcen zugeordnet sind, die Teil der Ressourcengruppe sind. Wenn Sie ein Dashboard mit dem Namen CloudWatch-Default- erstellt haben ResourceGroupName, wird es außerdem im Standard-Dashboard-Bereich angezeigt.

Sie können weiter nach unten gehen, indem Sie sich gleichzeitig auf einen einzelnen AWS Service und eine Ressourcengruppe konzentrieren. Die folgende Vorgehensweise zeigt, wie Sie sich auf eine Ressourcengruppe konzentrieren können.

So konzentrieren Sie sich auf eine einzelne Ressourcengruppe

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Oben auf der Seite, wo die All resources (Alle Ressourcen) angezeigt wird, wählen Sie eine Ressourcengruppe aus.
3. Um weitere Metriken zu dieser Ressourcengruppe am unteren Bildschirmrand anzuzeigen, wählen Sie View cross service dashboard (Serviceübergreifendes Dashboard anzeigen) aus.

Das serviceübergreifende Dashboard wird angezeigt und führt nur die Services auf, die sich auf diese Ressourcengruppe beziehen. Für jeden Service werden ein oder zwei Schlüsselmetriken angezeigt.

4. Um den Zeitraum zu ändern, der in allen aktuell angezeigten Diagrammen und Alarmen angezeigt wird, wählen Sie neben Time range (Zeitraum) oben auf dem Bildschirm den gewünschten Zeitraum aus. Um aus mehr Zeitraumsoptionen auszuwählen, als standardmäßig angezeigt werden, wählen Sie custom (benutzerdefiniert) aus.
5. Alarme werden immer einmal pro Minute aktualisiert. Um die Ansicht zu aktualisieren, wählen Sie das Aktualisierungssymbol (zwei gebogene Pfeile) oben rechts auf dem Bildschirm aus. Um die automatische Aktualisierungsrate für andere Elemente auf dem Bildschirm als Alarme zu ändern, wählen Sie den Pfeil nach unten neben dem Aktualisierungssymbol aus und klicken Sie auf eine Aktualisierungsrate. Sie können die automatische Aktualisierung auch deaktivieren.
6. Um zur Anzeige von Informationen über alle Ressourcen in Ihrem Konto zurückzukehren, wählen Sie oben auf dem Bildschirm, wo aktuell der Name der Ressourcengruppe angezeigt wird, All resources (Alle Ressourcen) aus.

CloudWatch Abrechnung und Kosten

In diesem Abschnitt wird beschrieben, wie CloudWatch Amazon-Funktionen Kosten verursachen. Es bietet auch Methoden, mit denen Sie Kosten analysieren, optimieren und CloudWatch senken können. In diesem Abschnitt beziehen wir uns bei der Beschreibung von CloudWatch Funktionen manchmal auf die Preisgestaltung. Informationen zu den Preisen finden Sie unter CloudWatch Amazon-Preise.

Themen

Analysieren Sie CloudWatch Kosten- und Nutzungsdaten mit dem Cost Explorer

Mit AWS Cost Explorer können Sie Kosten- und Nutzungsdaten für einen längeren AWS-Services Zeitraum visualisieren und analysieren, unter anderem CloudWatch. Weitere Informationen finden Sie unter Erste Schritte mit AWS Cost Explorer.

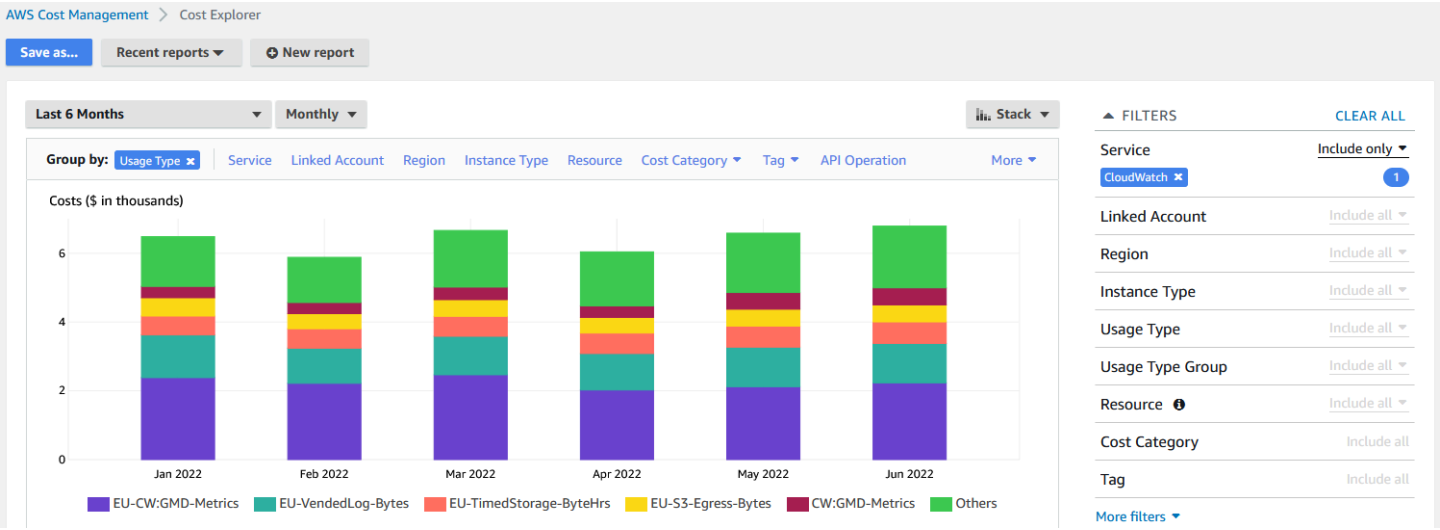
Das folgende Verfahren beschreibt, wie Sie Cost Explorer verwenden, um Kosten- und Nutzungsdaten zu visualisieren und zu analysieren CloudWatch .

Um CloudWatch Kosten- und Nutzungsdaten zu visualisieren und zu analysieren

1. Melden Sie sich unter https://console.aws.amazon.com/cost-management/home#/custom bei der Cost Explorer-Konsole an.
2. Wählen Sie unter FILTER für Service die Option aus CloudWatch.
3. Wählen Sie unter Group by (Gruppieren nach) die Option Usage Type (Nutzungstyp) aus. Sie können Ihre Ergebnisse auch nach anderen Kategorien gruppieren, beispielsweise:
   - API Operation (API-Betrieb): Zeigt, welche API-Operationen die meisten Kosten verursacht haben.

- Region: Zeigt, welche Regionen die meisten Kosten verursacht haben.

Die folgende Abbildung zeigt ein Beispiel für die Kosten, die CloudWatch Funktionen über einen Zeitraum von sechs Monaten verursacht haben.



Sehen Sie sich die Werte für an, um zu sehen, welche CloudWatch Funktionen die meisten Kosten verursacht habenUsageType. Stellt beispielsweise die Kosten `EU-CW:GMD-Metrics` dar, die durch CloudWatch Massen-API-Anfragen generiert wurden.

> **Note**
>
> Die Zeichenfolgen für `UsageType` entsprechen spezifischen Features und Regionen. Beispielsweise entspricht der erste Teil von `EU-CW:GMD-Metrics` (`EU`) der Region Europa (Irland) und der zweite Teil von `EU-CW:GMD-Metrics` (`GMD-Metrics`) entspricht CloudWatch API-Massenanfragen.
>
> Die gesamte Zeichenfolge für `UsageType` kann wie folgt formatiert sein: `<Region>-CW:<Feature>` oder `<Region>-<Feature>`.
>
> Zur besseren Lesbarkeit wurden die Zeichenfolgen für `UsageType` in den Tabellen dieses Dokuments auf ihre Zeichenfolgensuffixe reduziert. `EU-CW:GMD-Metrics` wurde beispielsweise zu `GMD-Metrics` verkürzt.

Die folgende Tabelle enthält die Namen der einzelnen CloudWatch Funktionen, listet die Namen der einzelnen Unterfunktionen auf und listet die Zeichenfolgen für `UsageType` auf.

CloudWatch Feature	CloudWatch Untermerkmal	**`UsageType`**
CloudWatch Metriken	Eigene Metriken	`MetricMonitorUsage`
	Detaillierte Überwachung	`MetricMonitorUsage`
	Eingebettete Metriken	`MetricMonitorUsage`
CloudWatch API-Anfragen	API-Anforderungen	`Requests`
	Massenvorgang (Abrufen)	`GMD-Metrics`
	Contributor Insights	`GIRR-Metrics`
	Bitmap-Bild (Snapshot)	`GMWI-Metrics`
CloudWatch metrische Streams	Metrik-Streams	`MetricStreamUsage`
CloudWatch Dashboards	Dashboard mit 50 oder weniger Metriken	`DashboardsUsageHour-Basic`
	Dashboard mit mehr als 50 Metriken	`DashboardsUsageHour`
CloudWatch Alarme	Standard-Metrikalarm (metrischer Alarm)	`AlarmMonitorUsage`
	Hochauflösend (Metrik-Alarm)	`HighResAlarmMonitorUsage`
	Metrics-Insights-Abfragealarm	

CloudWatch Feature	CloudWatch Untermerkmal	**UsageType**
		`MetricInsightAlarmUsage`
	Komposit (aggregierter Alarm)	`CompositeAlarmMonitorUsage`
CloudWatch Anwendungssignale	Anwendungssignale	`Application-Signals`
CloudWatch benutzerdefinierte Protokolle	Sammeln (Erfassen)	`DataProcessing-Bytes`
	Speichern (Archivieren)	`TimedStorage-ByteHrs`
	Analysieren (Abfragen)	`DataScanned-Bytes`
CloudWatch Seltene Zugriffsprotokolle	Sammeln (Erfassen)	`DataProcessingIA-Bytes`
CloudWatch verkaufte Protokolle	Lieferung (Amazon CloudWatch Logs)	`VendedLog-Bytes`
	Lieferung (CloudWatch protokolliert Protokolle für seltene Zugriffe)	`VendedLogIA-Bytes`
	Übermittlung (Amazon Simple Storage Service)	`S3-Egress-ComprBytes` `S3-Egress-Bytes`
	Lieferung (Amazon Data Firehose)	`FH-Egress-Bytes`

CloudWatch Feature	CloudWatch Untermerkmal	**`UsageType`**
Contributor Insights	CloudWatch Protokolle (Regeln)	`ContributorInsightRules`
	CloudWatch Protokolle (Ereignisse)	`ContributorInsightEvents`
	Amazon DynamoDB (Regeln)	`ContributorRulesManaged`
	DynamoDB (Ereignisse)	`ContributorEventsManaged`
Canarys (Synthetics)	Führen Sie	`Canary-runs`
Evidently	Ereignisse	`Evidently-event`
	Analyseeinheiten	`Evidently-eau`
RUM	Ereignisse	`RUM-event`

Analysieren Sie CloudWatch Kosten- und Nutzungsdaten mit AWS Cost and Usage Report s und Athena

Eine weitere Möglichkeit, CloudWatch Kosten- und Nutzungsdaten zu analysieren, ist die Verwendung von AWS Cost and Usage Report s mit Amazon Athena. AWS Cost and Usage Report s enthalten einen umfassenden Satz von Kosten- und Nutzungsdaten. Sie können Berichte erstellen, die Ihre Kosten und Ihre Nutzung nachverfolgen, und diese Berichte in einem S3-Bucket Ihrer Wahl veröffentlichen. Sie können Berichte auch aus Ihrem S3-Bucket herunterladen und löschen. Weitere

Informationen finden Sie unter Was sind AWS Cost and Usage Report s? im AWS Cost and Usage Report s-Benutzerhandbuch.

> **Note**
>
> Die Verwendung von AWS Cost and Usage Report s ist kostenlos. Sie zahlen nur für den Speicherplatz, wenn Sie Ihre Berichte in Amazon Simple Storage Service (Amazon S3) veröffentlichen. Weitere Informationen finden Sie im Benutzerhandbuch zu AWS Cost and Usage Report en unter Kontingente und Einschränkungen.

Athena ist ein Abfragedienst, den Sie mit AWS Cost and Usage Report s verwenden können, um Kosten- und Nutzungsdaten zu analysieren. Sie können Ihre Berichte in Ihrem S3-Bucket abfragen, ohne sie vorher herunterzuladen. Weitere Informationen finden Sie im Benutzerhandbuch zu Amazon Athena unter Was ist Amazon Athena?. Weitere Informationen finden Sie im Benutzerhandbuch zu Amazon Athena unter Was ist Amazon Athena?. Weitere Informationen zur Preisgestaltung finden Sie unter Amazon Athena – Preise.

Das folgende Verfahren beschreibt den Prozess zur Aktivierung von AWS Cost and Usage Report s und zur Integration des Dienstes in Athena. Das Verfahren enthält zwei Beispielabfragen, mit denen Sie CloudWatch Kosten- und Nutzungsdaten analysieren können.

> **Note**
>
> Sie können jede Beispielabfrage aus diesem Dokument verwenden. Alle Beispielabfragen in diesem Dokument werden für eine Datenbank namens costandusagereport ausgeführt und liefern Ergebnisse für den Monat April und das Jahr 2022. Diese Informationen können geändert werden. Vergewissern Sie sich jedoch vor dem Ausführen einer Abfrage, dass der Name Ihrer Datenbank mit dem Namen der Datenbank in der Abfrage übereinstimmt.

Um Kosten- und Nutzungsdaten mit AWS Cost and Usage Report s und Athena zu analysieren

1. Aktivieren Sie AWS Cost and Usage Report s. Weitere Informationen finden Sie im Benutzerhandbuch zu AWS Cost and Usage Report en unter Erstellen von Kosten- und Nutzungsberichten.

> **Tip**
>
> Wählen Sie beim Erstellen Ihrer Berichte die Option Include resource IDs (Ressourcen-IDs einschließen) aus. Andernfalls enthalten Ihre Berichte die Spalte `line_item_resource_id` nicht. Anhand dieser Zeile lassen sich Kosten bei der Analyse von Kosten- und Nutzungsdaten näher identifizieren.

2. Integrieren Sie AWS Cost and Usage Report uns in Athena. Weitere Informationen finden Sie unter Athena mithilfe von AWS CloudFormation Vorlagen einrichten im AWS Cost and Usage Report s User Guide.
3. Fragen Sie Ihre Kosten- und Nutzungsberichte ab.

Beispiel: Athena-Abfrage

Mithilfe der folgenden Abfrage können Sie ermitteln, welche CloudWatch Funktionen in einem bestimmten Monat die meisten Kosten verursacht haben.

```
SELECT
CASE
-- Metrics
WHEN line_item_usage_type LIKE '%%MetricMonitorUsage%%' THEN 'Metrics (Custom, Detailed
 monitoring management portal EMF)'
WHEN line_item_usage_type LIKE '%%Requests%%' THEN 'Metrics (API Requests)'
WHEN line_item_usage_type LIKE '%%GMD-Metrics%%' THEN 'Metrics (Bulk API Requests)'
WHEN line_item_usage_type LIKE '%%MetricStreamUsage%%' THEN 'Metric Streams'
-- Dashboard
WHEN line_item_usage_type LIKE '%%DashboardsUsageHour%%' THEN 'Dashboards'
-- Alarms
WHEN line_item_usage_type LIKE '%%AlarmMonitorUsage%%' THEN 'Alarms (Standard)'
WHEN line_item_usage_type LIKE '%%HighResAlarmMonitorUsage%%' THEN 'Alarms (High
 Resolution)'
WHEN line_item_usage_type LIKE '%%MetricInsightAlarmUsage%%' THEN 'Alarms (Metrics
 Insights)'
WHEN line_item_usage_type LIKE '%%CompositeAlarmMonitorUsage%%' THEN 'Alarms
 (Composite)'
-- Logs
WHEN line_item_usage_type LIKE '%%DataProcessing-Bytes%%' THEN 'Logs (Collect - Data
 Ingestion)'
-- Logs
```

```
WHEN line_item_usage_type LIKE '%%DataProcessingIA-Bytes%%' THEN 'Infrequent Access
 Logs (Collect - Data Ingestion)'
WHEN line_item_usage_type LIKE '%%TimedStorage-ByteHrs%%' THEN 'Logs (Storage -
 Archival)'
WHEN line_item_usage_type LIKE '%%DataScanned-Bytes%%' THEN 'Logs (Analyze - Logs
 Insights queries)'
-- Vended Logs
WHEN line_item_usage_type LIKE '%%VendedLog-Bytes%%' THEN 'Vended Logs (Delivered to
 CW)'
WHEN line_item_usage_type LIKE '%%VendedLogIA-Bytes%%' THEN 'Vended Infrequent Access
 Logs (Delivered to CW)'
WHEN line_item_usage_type LIKE '%%FH-Egress-Bytes%%' THEN 'Vended Logs (Delivered to
 Kinesis FH)'
WHEN (line_item_usage_type LIKE '%%S3-Egress-Bytes%%') OR (line_item_usage_type LIKE '%
%S3-Egress-
ComprBytes%%') THEN 'Vended Logs (Delivered to S3)'
-- Other
WHEN line_item_usage_type LIKE '%%Application-Signals%%' THEN 'Application Signals'
WHEN line_item_usage_type LIKE '%%Canary-runs%%' THEN 'Synthetics'
WHEN line_item_usage_type LIKE '%%Evidently%%' THEN 'Evidently'
WHEN line_item_usage_type LIKE '%%RUM-event%%' THEN 'RUM'
ELSE 'Others'
END AS UsageType,
-- REGEXP_EXTRACT(line_item_resource_id,'^(?:.+?:){5}(.+)$',1) as ResourceID,
-- SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
-- AND line_item_usage_account_id = '123456789012' - If you want to filter on a
 specific account, you can
remove this comment at the beginning of the line and specify an AWS account.
GROUP BY
1
ORDER BY
TotalSpend DESC,
UsageType;
```

Beispiel: Athena-Abfrage

Mit der folgenden Abfrage können Sie die Ergebnisse für `UsageType` und `Operation` anzeigen. Dies zeigt Ihnen, wie CloudWatch Funktionen Kosten verursacht haben. Die Ergebnisse zeigen auch die Werte für `UsageQuantity` und `TotalSpend`, sodass Sie Ihre Gesamtnutzungskosten sehen können.

> **Tip**
>
> Wenn Sie weitere Informationen zu `UsageType` erhalten möchten, können Sie der Abfrage die folgende Zeile hinzufügen:
> `line_item_line_item_description`
> Diese Zeile erstellt eine Spalte mit dem Namen Description (Beschreibung).

```
SELECT
bill_payer_account_id as Payer,
line_item_usage_account_id as LinkedAccount,
line_item_usage_type AS UsageType,
line_item_operation AS Operation,
line_item_resource_id AS ResourceID,
SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
GROUP BY
bill_payer_account_id,
line_item_usage_account_id,
line_item_usage_type,
line_item_resource_id,
line_item_operation
```

Bewährte Methoden zur Kostenoptimierung und -senkung

CloudWatch Metriken

Viele AWS-Services, wie Amazon Elastic Compute Cloud (Amazon EC2), Amazon S3 und Amazon Data Firehose, senden automatisch und kostenlos Metriken CloudWatch an. Für Metriken aus den folgenden Kategorien fallen jedoch unter Umständen zusätzliche Kosten an:

- Eigene Metriken, detaillierte Überwachung und eingebettete Metriken
- API-Anforderungen
- Metrik-Streams

Weitere Informationen finden Sie unter Verwenden von CloudWatch Amazon-Metriken.

Eigene Metriken, detaillierte Überwachung und eingebettete Metriken

Benutzerdefinierte Metriken

Sie können eigene Metriken erstellen, um Datenpunkte in beliebiger Reihenfolge und Geschwindigkeit zu strukturieren.

Alle eigenen Metriken werden anteilsmäßig pro Stunde berechnet. Sie werden nur gemessen, wenn sie an CloudWatch gesendet werden. Informationen zur Preisgestaltung von Kennzahlen finden Sie unter CloudWatch Amazon-Preise.

In der folgenden Tabelle sind die Namen der relevanten Unterfunktionen für CloudWatch Metriken aufgeführt. Die Tabelle enthält auch die Zeichenfolgen für `UsageType` und `Operation`, die bei der Analyse und Identifizierung metrikbezogener Kosten hilfreich sein können.

> **Note**
>
> Wenn Sie beim Abfragen von Kosten- und Nutzungsdaten mit Athena weitere Details zu den Metriken aus der folgenden Tabelle erhalten möchten, gleichen Sie die Zeichenfolgen für `Operation` mit den Ergebnissen ab, die für `line_item_operation` angezeigt werden.

CloudWatchUnterfunktion	**`UsageType`**	**`Operation`**	Zweck

CloudWatchUnterfunktion	**UsageType**	**Operation**	Zweck
Eigene Metriken	`MetricMonitorUsage`	`MetricStorage`	Eigene Metriken
Detaillierte Überwachung	`MetricMonitorUsage`	`MetricStorage:AWS/`*`{Service}`*	Detaillierte Überwachung
Eingebettete Metriken	`MetricMonitorUsage`	`MetricStorage:AWS/Logs-EMF`	Protokolliert eingebettete Metriken
Protokollfilter	`MetricMonitorUsage`	`MetricStorage:AWS/CloudWatchLogs`	Metrikfilter für Protokollgruppen

Detaillierte Überwachung

CloudWatch hat zwei Arten der Überwachung:

- Grundlegende Überwachung

  Die Grundlegende Überwachung ist kostenlos und wird automatisch für alle AWS-Services aktiviert, die das Feature unterstützen.

- Detaillierte Überwachung

  Eine detaillierte Überwachung ist mit Kosten verbunden und bietet je nach Bedarf unterschiedliche Verbesserungen. AWS-Service Bei jedem AWS-Service , der die detaillierte Überwachung unterstützt, können Sie wählen, ob die detaillierte Überwachung für den jeweiligen Service aktiviert werden soll. Weitere Informationen finden Sie unter Grundlegende Überwachung und detaillierte Überwachung.

> **Note**
>
> Andere AWS-Services unterstützen eine detaillierte Überwachung und verweisen möglicherweise unter einem anderen Namen auf diese Funktion. Bei Amazon S3 wird die detaillierte Überwachung beispielsweise als Anforderungsmetriken bezeichnet.

Ähnlich wie bei benutzerdefinierten Messwerten erfolgt die detaillierte Überwachung anteilig pro Stunde und nur dann, wenn Daten gesendet werden. CloudWatch Eine detaillierte Überwachung verursacht Kosten, die sich nach der Anzahl der Metriken richten, an die gesendet werden. CloudWatch Aus Kostengründen sollte die detaillierte Überwachung nur bei Bedarf aktiviert werden. Informationen zu den Preisen für detailliertes Monitoring finden Sie unter CloudWatch Amazon-Preise.

Beispiel: Athena-Abfrage

Mit der folgende Abfrage können Sie prüfen, für welche EC2-Instances die detaillierte Überwachung aktiviert ist.

```
SELECT
bill_payer_account_id as Payer,
line_item_usage_account_id as LinkedAccount,
line_item_usage_type AS UsageType,
line_item_operation AS Operation,
line_item_resource_id AS ResourceID,
SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_operation='MetricStorage:AWS/EC2'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
GROUP BY
bill_payer_account_id,
line_item_usage_account_id,
line_item_usage_type,
line_item_resource_id,
```

```
line_item_operation,
line_item_line_item_description
ORDER BY line_item_operation
```

Eingebettete Metriken

Mit dem CloudWatch eingebetteten Metrikformat können Sie Anwendungsdaten als Protokolldaten aufnehmen, sodass Sie verwertbare Metriken generieren können. Weitere Informationen finden Sie unter Erfassung von Protokollen mit hoher Kardinalität und Generieren von Metriken mit dem eingebetteten Metrikformat. CloudWatch

Eingebettete Metriken verursachen Kosten. Diese basieren auf der Anzahl der erfassten Protokolle, der Anzahl der archivierten Protokolle und der Anzahl der generierten eigenen Metriken.

In der folgenden Tabelle sind die Namen der relevanten Unterfunktionen für das eingebettete metrische Format aufgeführt. CloudWatch Die Tabelle enthält auch die Zeichenfolgen für `UsageType` und `Operation`, die bei der Analyse und Identifizierung der Kosten hilfreich sein können.

CloudWatch Unterfunktion	**`UsageType`**	**`Operation`**	Zweck
Eigene Metriken	`MetricMonitorUsage`	`MetricStorage:AWS/Logs-EMF`	Protokolliert eingebettete Metriken
Erfassung von Protokollen	`DataProcessing-Bytes`	`PutLogEvents`	Lädt einen Batch von Protokollereignissen in die angegebene Protokollgruppe oder in den angegebenen Protokollstream hoch
Archivierung von Protokollen	`TimedStorage-ByteHrs`	`HourlyStorageMetering`	Speichert Protokolle pro Stunde und Protokolle pro Byte in CloudWatch Logs

Um Kosten zu analysieren, verwenden Sie AWS Cost and Usage Report s mit Athena, damit Sie ermitteln können, welche Kennzahlen Kosten verursachen, und bestimmen können, wie die Kosten generiert werden.

Um die durch das CloudWatch eingebettete Metrikformat generierten Kosten optimal zu nutzen, sollten Sie es vermeiden, Kennzahlen zu erstellen, die auf Dimensionen mit hoher Kardinalität basieren. Auf diese Weise wird CloudWatch nicht für jede einzelne Dimensionskombination eine benutzerdefinierte Metrik erstellt. Weitere Informationen finden Sie unter Dimensionen.

Wenn Sie CloudWatch Container Insights verwenden, um das eingebettete Metrikformat zu nutzen, können Sie AWS Distro for Open Telemetry als Alternative verwenden, um das Beste aus den Kosten für Kennzahlen herauszuholen. Mit Container Insights können Sie Metriken und Protokolle aus Ihren containerisierten Anwendungen und Microservices sammeln, aggregieren und zusammenfassen. Wenn Sie Container Insights aktivieren, sendet der CloudWatch Agent Ihre Logs an CloudWatch, sodass er anhand der Logs eingebettete Metriken generieren kann. Der CloudWatch Agent sendet jedoch nur eine feste Anzahl von Metriken an CloudWatch, und Ihnen werden alle verfügbaren Metriken in Rechnung gestellt, auch solche, die Sie nicht verwenden. Mit AWS Distro for Open Telemetry können Sie konfigurieren und anpassen, an welche Metriken und Dimensionen gesendet werden. CloudWatch Dadurch können Sie das Datenvolumen und die Kosten reduzieren, die durch Container Insights entstehen. Weitere Informationen finden Sie in den folgenden Ressourcen:

- Verwenden von Container Insights
- AWS Distro für Open Telemetry

API-Anforderungen

CloudWatch hat die folgenden Arten von API-Anfragen:

- API-Anforderungen
- Massenvorgang (Abrufen)
- Contributor Insights
- Bitmap-Bild (Snapshot)

API-Anforderungen verursachen Kosten. Diese basieren auf dem Anforderungstyp und auf der Anzahl der angeforderten Metriken.

Die Tabelle enthält die Arten von API-Anforderungen sowie die Zeichenfolgen für `UsageType` und `Operation`, die bei der Analyse und Identifizierung API-bezogener Kosten hilfreich sein können.

API-Anforderungstyp	**`UsageType`**	**`Operation`**	Zweck
API-Anforderungen	`Requests`	`GetMetricStatistics`	Ruft Statistiken für die angegebenen Metriken ab
	`Requests`	`ListMetrics`	Listet die angegebenen Metriken auf
	`Requests`	`PutMetricData`	Veröffentlicht metrische Datenpunkte in CloudWatch
	`Requests`	`GetDashboard`	Zeigt Details für die angegebenen Dashboards an
	`Requests`	`ListDashboards`	Listet die Dashboards in Ihrem Konto auf
	`Requests`	`PutDashboard`	Erstellt oder aktualisiert ein Dashboard
	`Requests`	`DeleteDashboards`	Löscht alle angegebenen Dashboards
Massenvorgang (Abrufen)	`GMD-Metrics`	`GetMetricData`	Ruft CloudWatch metrische Werte ab
Contributor Insights	`GIRR-Metrics`	`GetInsightRuleReport`	Gibt Zeitreihendaten zurück, die durch eine Contributor-Insights-Regel gesammelt wurden

API-Anforderungstyp	**`UsageType`**	**`Operation`**	Zweck
Bitmap-Bild (Snapshot)	`GMWI-Metrics`	`GetMetricWidgetImage`	Ruft eine Momentaufnahme einer oder mehrerer CloudWatch Metriken als Bitmap-Bild ab

Verwenden Sie Cost Explorer für die Kostenanalyse und gruppieren Sie Ihre Ergebnisse nach API Operation (API-Betrieb).

Die Kosten für API-Anfragen variieren, und es fallen Kosten an, wenn Sie die Anzahl der API-Aufrufe überschreiten, die Ihnen im Rahmen des AWS kostenlosen Kontingents zur Verfügung gestellt wurden.

> **Note**
>
> `GetMetricData`und `GetMetricWidgetImage` sind nicht im Limit des AWS kostenlosen Kontingents enthalten. Weitere Informationen finden Sie im AWS Billing Benutzerhandbuch unter Nutzung des AWS kostenlosen Kontingents.

Die API-Anforderungen, die üblicherweise Kosten verursachen, sind Abfragen vom Typ `Put` und `Get`.

`PutMetricData`

`PutMetricData` verursacht bei jedem Aufruf Kosten. Diese können je nach Anwendungsfall erheblich sein. Weitere Informationen finden Sie PutMetricDatain der Amazon CloudWatch API-Referenz.

Fassen Sie für eine möglichst kosteneffiziente Nutzung von `PutMetricData` mehr Daten in Ihren API-Aufrufen zu einem Batch zusammen. Abhängig von Ihrem Anwendungsfall sollten Sie erwägen, CloudWatch Logs oder das CloudWatch eingebettete Metrikformat zum Einfügen von Metrikdaten zu verwenden. Weitere Informationen finden Sie in den folgenden Ressourcen:

- Was ist Amazon CloudWatch Logs? im Amazon CloudWatch Logs-Benutzerhandbuch
- Erfassung von Protokollen mit hoher Kardinalität und Generierung von Metriken mit eingebettetem Metrikformat CloudWatch

- Senkung der Kosten und Fokussierung auf unsere Kunden mit CloudWatch integrierten benutzerdefinierten Kennzahlen von Amazon

GetMetricData

`GetMetricData` kann ebenfalls erhebliche Kosten verursachen. Zu den gängigsten kostentreibenden Anwendungsfällen gehören Überwachungstools von Drittanbietern, die Daten abrufen, um Erkenntnisse zu generieren. Weitere Informationen finden Sie GetMetricDatain der Amazon CloudWatch API-Referenz.

Zur Senkung der durch `GetMetricData` entstehenden Kosten empfiehlt es sich gegebenenfalls, nur Daten abzurufen, die überwacht und verwendet werden, oder Daten seltener abzurufen. Je nach Anwendungsfall können Sie ggf. Metrik-Streams anstelle von `GetMetricData` verwenden, um Daten nahezu in Echtzeit und zu geringeren Kosten an Dritte zu pushen. Weitere Informationen finden Sie in den folgenden Ressourcen:

- Verwenden von Streaming-Metriken
- CloudWatch Metric Streams — Senden Sie AWS Metriken in Echtzeit an Partner und an Ihre Apps

GetMetricStatistics

Je nach Anwendungsfall können Sie ggf. `GetMetricStatistics` anstelle von `GetMetricData` verwenden. Mit `GetMetricData` können Sie Daten schnell und im großen Stil abrufen. Es `GetMetricStatistics` ist jedoch im Rahmen des AWS kostenlosen Kontingents für bis zu eine Million API-Anfragen enthalten. Dies kann Ihnen helfen, die Kosten zu senken, wenn Sie nicht so viele Metriken und Datenpunkte pro Anruf abrufen müssen. Weitere Informationen finden Sie in den folgenden Ressourcen:

- GetMetricStatisticsin der Amazon CloudWatch API-Referenz
- Sollte ich GetMetricData oder verwenden GetMetricStatistics?

> **Note**
>
> Externe Aufrufer führen API-Aufrufe aus. Derzeit besteht die einzige Möglichkeit, diese Anrufer zu identifizieren, darin, eine Anfrage an den technischen Support an das CloudWatch Team zu richten und nach Informationen über sie zu fragen. Informationen zum Erstellen

einer Anfrage an technischen Support finden Sie unter Wie erhalte ich technischen Support von AWS? .

CloudWatch metrische Ströme

Mit CloudWatch metrischen Streams können Sie Metriken kontinuierlich an AWS Ziele und Ziele von Drittanbietern senden.

Die durch Metrik-Streams entstehenden Kosten basieren auf der Anzahl von Metrikaktualisierungen. Metrikaktualisierungen enthalten immer Werte für die folgenden Statistiken:

- `Minimum`
- `Maximum`
- `Sample Count`
- `Sum`

Weitere Informationen finden Sie unter Statistiken, die gestreamt werden können.

Verwenden Sie AWS Cost and Usage Report s mit Athena, um Kosten zu analysieren, die durch CloudWatch metrische Streams generiert werden. Dadurch können Sie identifizieren, welche Metrik-Streams Kosten verursachen, und ermitteln, wie die Kosten entstehen.

Beispiel: Athena-Abfrage

Mit der folgenden Abfrage können Sie nachverfolgen, welche Metrik-Streams Kosten verursachen – aufgeschlüsselt nach Amazon-Ressourcenname (ARN):

```
SELECT
SPLIT_PART(line_item_resource_id,'/',2) AS "Stream Name",
line_item_resource_id as ARN,
SUM(CAST(line_item_unblended_cost AS decimal(16,2))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
```

```
-- AND line_item_usage_account_id = '123456789012' - If you want to filter on a
 specific account, you can
remove this comment at the beginning of the line and specify an AWS account.
AND line_item_usage_type LIKE '%%MetricStreamUsage%%'
GROUP BY line_item_resource_id
ORDER BY TotalSpend DESC
```

Um die durch CloudWatch metrische Streams verursachten Kosten zu reduzieren, streamen Sie nur die Kennzahlen, die Ihrem Unternehmen einen Mehrwert bieten. Sie können auch jeden ungenutzten Metrik-Stream beenden oder anhalten.

CloudWatch Alarme

Mit CloudWatch Alarmen können Sie Alarme erstellen, die auf einer einzelnen Metrik basieren, Alarme, die auf einer Metrics Insights-Abfrage basieren, und zusammengesetzte Alarme erstellen, die andere Alarme beobachten.

> **Note**
>
> Die Kosten für metrische und zusammengesetzte Alarme werden anteilsmäßig pro Stunde berechnet. Kosten für Ihre Alarme entstehen Ihnen nur, solange Ihre Alarme existieren. Um die Kosten zu optimieren, sollten Sie darauf achten, keine falsch konfigurierten oder minderwertigen Alarme zu vergessen. Um Ihnen dabei zu helfen, können Sie die Bereinigung von CloudWatch Alarmen, die Sie nicht mehr benötigen, automatisieren. Weitere Informationen finden Sie unter Automatisieren von Amazon CloudWatch Alarm Cleanup at Scale

Metric alarms (Metrikalarme)

Metrikalarme haben folgende Auflösungseinstellungen:

- Standard (Auswertung alle 60 Sekunden)
- High resolution (Hohe Auflösung; Auswertung alle 10 Sekunden)

Wenn Sie einen metrischen Alarm erstellen, basieren Ihre Kosten auf der Auflösungseinstellung Ihres Alarms und der Anzahl der Metriken, auf die Ihr Alarm verweist. Beispielsweise verursacht ein metrischer Alarm, der auf eine Metrik verweist, eine Alarmmetrik pro Stunde. Weitere Informationen finden Sie unter CloudWatch Amazon-Alarme verwenden.

Wenn Sie einen Metrikalarm erstellen, der einen mathematischen Metrikausdruck enthält, der auf mehrere Metriken verweist, entstehen Kosten für jede Alarmmetrik, auf die im mathematischen Metrikausdruck verwiesen wird. Informationen zum Erstellen eines metrischen Alarms, der einen metrischen mathematischen Ausdruck enthält, finden Sie unter Einen CloudWatch Alarm auf der Grundlage eines metrischen mathematischen Ausdrucks erstellen.

Wenn Sie einen Anomalieerkennungsalarm erstellen, bei dem Ihr Alarm vergangene Metrikdaten analysiert, um ein Modell mit erwarteten Werten zu erstellen, entstehen Kosten für jede Alarm-Metrik, auf die in Ihrem Alarm verwiesen wird, sowie für zwei zusätzliche Metriken, eine für die obere und untere Bandmetrik, die das Anomalieerkennungsmodell erstellt. Informationen zum Erstellen eines Alarms bei der Erkennung von Anomalien finden Sie unter Erstellen eines CloudWatch Alarms auf der Grundlage der Anomalieerkennung.

Metrics-Insights-Abfragealarme

Bei Metric-Insights-Abfragealarmen handelt es sich um eine bestimmte Art von metrischem Alarm, der nur mit Standardauflösung verfügbar ist (alle 60 Sekunden ausgewertet).

Wenn Sie einen Metric-Insights-Abfragealarm erstellen, basieren Ihre Kosten auf der Anzahl der Metriken, die von der Abfrage analysiert werden, auf die sich Ihr Alarm bezieht. Beispielsweise verursacht ein Metric-Insights-Abfragealarm, der auf eine Abfrage verweist, deren Filter zehn Metriken entspricht, zehn analysierte Metriken pro Stunde. Weitere Informationen finden Sie im Preisbeispiel auf Amazon CloudWatch Pricing.

Wenn Sie einen Alarm erstellen, der sowohl eine Metrics-Insights-Abfrage als auch einen metrischen mathematischen Ausdruck enthält, wird er als Metrics-Insights-Abfragealarm gemeldet. Wenn Ihr Alarm einen metrischen mathematischen Ausdruck enthält, der zusätzlich zu den von der Metrics-Insights-Abfrage analysierten Metriken auf andere Metriken verweist, entstehen Ihnen zusätzliche Kosten für jede Alarm-Metrik, auf die im metrischen mathematischen Ausdruck verwiesen wird. Informationen zum Erstellen eines metrischen Alarms, der einen metrischen mathematischen Ausdruck enthält, finden Sie unter Erstellen eines CloudWatch Alarms auf der Grundlage eines metrischen mathematischen Ausdrucks.

Zusammengesetzte Alarme (Composite alarms)

Zusammengesetzte Alarme enthalten Regelausdrücke, die angeben, wie sie die Zustände anderer Alarme auswerten sollen, um ihre eigenen Zustände zu bestimmen. Bei zusammengesetzten Alarmen fallen Standardkosten pro Stunde an, unabhängig davon, wie viele andere Alarme sie auswerten. Alarme, auf die sich zusammengesetzte Alarme in Regelausdrücken beziehen,

verursachen separate Kosten. Weitere Informationen finden Sie unter Erstellen eines zusammengesetzten Alarms.

Alarm usage types (Alarm-Einsatztypen)

In der folgenden Tabelle sind die Namen der relevanten Unterfunktionen für CloudWatch Alarme aufgeführt. Die Tabelle enthält die Zeichenfolgen für `UsageType`, die bei der Analyse und Identifizierung alarmbezogener Kosten hilfreich sein können.

CloudWatchUnterfunktion	**`UsageType`**
Standard-Metrikalarm	`AlarmMonitorUsage`
Hochauflösender metrischer Alarm	`HighResAlarmMonitorUsage`
Metrics-Insights-Abfragealarm	`MetricInsightAlarmUsage`
Zusammengesetzter Alarm	`CompositeAlarmMonitorUsage`

Reducing alarm costs (Reduzieren von Alarmkosten)

Um die Kosten zu optimieren, die durch mathematische Alarme mit vier oder mehr Kennzahlen generiert werden, können Sie Daten aggregieren, bevor sie an CloudWatch gesendet werden. Dadurch können Sie einen Alarm für eine einzelne Metrik erstellen, anstatt einen Alarm, der Daten für mehrere Metriken aggregiert. Weitere Informationen finden Sie unter Veröffentlichen benutzerdefinierter Metriken.

Um die durch Abfragealarme von Metrics Insights generierten Kosten zu optimieren, können Sie sicherstellen, dass der für die Abfrage verwendete Filter nur mit den Metriken übereinstimmt, die Sie überwachen möchten.

Die beste Methode zur Kostensenkung besteht darin, alle unnötigen oder ungenutzten Alarme zu entfernen. Sie können beispielsweise Alarme löschen, die Metriken auswerten, die von AWS Ressourcen ausgegeben werden, die nicht mehr existieren.

Beispiel: Überprüfen auf Alarme im ***`INSUFFICIENT_DATA`***-Status mit ***`DescribeAlarms`***

Wenn Sie eine Ressource löschen, aber nicht die Metrikalarme, die von der Ressource ausgegeben werden, sind die Alarme weiterhin vorhanden und werden normalerweise im `INSUFFICIENT_DATA`-

Zustand. Verwenden Sie den folgenden Befehl AWS Command Line Interface (AWS CLI), um nach Alarmen zu suchen, die sich im `INSUFFICIENT_DATA` Status befinden.

```
$ aws cloudwatch describe-alarms –state-value INSUFFICIENT_DATA
```

Weitere Möglichkeiten zur Kostensenkung:

- Achten Sie darauf, dass Sie Alarme für die richtigen Metriken erstellen.
- Achten Sie darauf, dass keine Alarme in Regionen aktiviert sind, in denen Sie nicht arbeiten.
- Denken Sie daran, dass zusammengesetzte Alarme zwar Geräusche reduzieren, aber auch zusätzliche Kosten verursachen.
- Berücksichtigen Sie bei der Entscheidung, welche Art von Alarm (Standardalarm oder hochauflösender Alarm) Sie erstellen, Ihren Anwendungsfall und den Nutzen des jeweiligen Alarmtyps.

CloudWatch Logs

Amazon CloudWatch Logs hat die folgenden Protokolltypen:

- Benutzerdefinierte Protokolle (Protokolle, die Sie für Ihre Anwendungen erstellen)
- Versendete Protokolle (Protokolle, die andere AWS-Services, wie Amazon Virtual Private Cloud (Amazon VPC) und Amazon Route 53, in Ihrem Namen erstellen)

Weitere Informationen zu verkauften Protokollen finden Sie unter Aktivieren der Protokollierung für bestimmte AWS Dienste im Amazon CloudWatch Logs-Benutzerhandbuch.

Die durch benutzerdefinierte Protokolle und Vended-Protokolle entstehenden Kosten basieren auf der Anzahl der gesammelten, gespeicherten und analysierten Protokolle. Unabhängig davon verursachen verkaufte Logs Kosten für die Lieferung an Amazon S3 und Firehose.

In der folgenden Tabelle sind die Namen der CloudWatch Logs-Funktionen und die Namen der relevanten Unterfunktionen aufgeführt. Die Tabelle enthält auch die Zeichenfolgen für `UsageType` und `Operation`, die bei der Analyse und Identifizierung protokollbezogener Kosten hilfreich sein können.

CloudWatch Funktion „Protokolle“	CloudWatch Unterfunktion „Protokolle“	**`UsageType`**	**`Operation`**	Zweck
Custom logs (Benutzer definierte Protokolle)	Sammeln (Erfassen)	`DataProcessing-Bytes`	`PutLogEvents`	Lädt einen Batch von Protokollen in einen bestimmten Protokollstream hoch
	Speichern (Archivieren)	`TimedStorage-ByteHrs`	`HourlyStorageMetering`	Speichert Protokolle pro Stunde und Protokolle pro Byte in CloudWatch Logs
	Analysieren (Logs-Insights-Abfragen)	`DataScanned-Bytes`	`StartQuery`	Protokolliert Daten, die durch CloudWatch Logs Insight-Abfragen gescannt wurden
Vended-Protokolle	Lieferung (CloudWatch Protokolle)	`VendedLog-Bytes`	`PutLogEvents`	Lädt einen Batch von Protokollen in einen bestimmten Protokollstream hoch
	Bereitstellung (Amazon S3)	`S3-Egress-ComprBytes`	`LogDelivery`	Sendet verkaufte Protokolle (CloudWat

CloudWatch Funktion „Protokolle“	CloudWatch Unterfunktion „Protokolle“	**UsageType**	**Operation**	Zweck
		S3-Egress-Bytes		chAmazon S3 oder Firehose)
	Lieferung (Firehose)	FH-Egress-Bytes	LogDelivery	Sendet verkaufte Protokolle (CloudWatchAmazon S3 oder Firehose)

Um Kosten zu analysieren, verwenden Sie AWS Cost and Usage Report s mit Athena, sodass Sie ermitteln können, welche Protokolle Kosten verursachen, und ermitteln können, wie die Kosten generiert werden.

Beispiel: Athena-Abfrage

Mit der folgenden Abfrage können Sie nachverfolgen, welche Protokolle Kosten verursachen – aufgeschlüsselt nach Ressourcen-ID:

```
SELECT
bill_payer_account_id as Payer,
line_item_usage_account_id as LinkedAccount,
line_item_resource_id AS ResourceID,
line_item_usage_type AS UsageType,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend,
SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_operation IN
 ('PutLogEvents','HourlyStorageMetering','StartQuery','LogDelivery')
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
```

```
GROUP BY
bill_payer_account_id,
line_item_usage_account_id,
line_item_usage_type,
line_item_resource_id,
line_item_operation
ORDER BY
TotalSpend DESC
```

Um das Beste aus den durch CloudWatch Logs generierten Kosten herauszuholen, sollten Sie Folgendes berücksichtigen:

- Protokollieren Sie nur Ereignisse, die einen Nutzen für Ihr Unternehmen haben. Dadurch fallen weniger Kosten für die Erfassung an.
- Ändern Sie Ihre Einstellungen für die Aufbewahrung von Protokollen, um die Kosten für die Speicherung zu senken. Weitere Informationen finden Sie unter Ändern der Aufbewahrung von Protokolldaten in CloudWatch Logs im Amazon CloudWatch Logs-Benutzerhandbuch.
- Führen Sie Abfragen aus, die CloudWatch Logs Insights automatisch in Ihrem Verlauf speichert. Dadurch fallen weniger Kosten für die Analyse an. Weitere Informationen finden Sie unter Laufende Abfragen oder Abfrageverlauf anzeigen im Amazon CloudWatch Logs-Benutzerhandbuch.
- Verwenden Sie den CloudWatch Agenten, um System- und Anwendungsprotokolle zu sammeln und an diese zu senden CloudWatch. Dies ermöglicht es, nur die Protokollereignisse zu erfassen, die Ihre Kriterien erfüllen. Weitere Informationen finden Sie unter Amazon CloudWatch Agent fügt Support für Protokollfilterausdrücke hinzu.

Um die Kosten für verkaufte Protokolle zu senken, sollten Sie Ihren Anwendungsfall berücksichtigen und dann entscheiden, ob Ihre Protokolle an Amazon S3 CloudWatch oder Amazon S3 gesendet werden sollen. Weitere Informationen finden Sie unter An Amazon S3 gesendete Logs im Amazon CloudWatch Logs-Benutzerhandbuch.

> **Tip**
>
> Wenn Sie Metrikfilter, Abonnementfilter, CloudWatch Logs Insights und Contributor Insights verwenden möchten, senden Sie verkaufte Logs an. CloudWatch
> Wenn Sie dagegen mit VPC-Flow-Protokollen arbeiten und diese für Überprüfungs- und Compliancezwecke verwenden, senden Sie Vended-Protokolle an Amazon S3.
> Informationen zur Nachverfolgung von Gebühren, die durch die Veröffentlichung von VPC Flow Logs in S3-Buckets generiert werden, finden Sie unter Verwenden von AWS Cost and

Usage Report s- und Kostenzuweisungs-Tags, um die Datenaufnahme von VPC FLOw Logs in Amazon S3 zu verstehen.

Weitere Informationen darüber, wie Sie die durch CloudWatch Logs generierten Kosten optimal nutzen können, finden Sie unter Welche Protokollgruppe verursacht einen plötzlichen Anstieg meiner Logs-Rechnung? CloudWatch .

CloudWatch Amazon-Dashboards verwenden

CloudWatch Amazon-Dashboards sind anpassbare Homepages in der CloudWatch Konsole, mit denen Sie Ihre Ressourcen in einer einzigen Ansicht überwachen können, auch die Ressourcen, die über verschiedene Regionen verteilt sind. Sie können CloudWatch Dashboards verwenden, um individuelle Ansichten der Kennzahlen und Alarme für Ihre AWS Ressourcen zu erstellen.

Mit Dashboards können Sie Folgendes erstellen:

- Eine einzige Ansicht für ausgewählte Metriken und Alarme, um Ihnen die Bewertung des Zustands Ihrer Ressourcen und Anwendungen in einer oder mehreren Regionen zu erleichtern. Sie können die für jede Metrik in jedem Diagramm verwendete Farbe auswählen, sodass Sie dieselbe Metrik über mehrere Diagramme hinweg mühelos verfolgen können.
- Ein operatives Playbook, das Teammitgliedern bei operativen Ereignissen Orientierungshilfen dazu bietet, wie auf bestimmte Vorfälle zu reagieren ist.
- Eine gemeinsame Ansicht wichtiger Maßnahmen für Ressourcen und Anwendungen, die von Teammitgliedern für einen schnelleren Kommunikationsfluss bei operativen Ereignissen gemeinsam genutzt wird.

Wenn Sie mehrere AWS Konten haben, können Sie die CloudWatch kontenübergreifende Beobachtbarkeit einrichten und anschließend umfangreiche kontenübergreifende Dashboards in Ihren Monitoring-Konten erstellen. Diese Dashboards können Diagramme mit Kennzahlen aus Quellkonten und CloudWatch Logs Insights-Widgets mit Abfragen von Protokollgruppen aus Quellkonten enthalten. Darüber hinaus können Alarme, die Sie im Überwachungskonto erstellen, Metriken in Quellkonten überwachen. Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.

Sie können Dashboards von der Konsole aus oder mithilfe der `PutDashboard` API-Operation AWS CLI oder erstellen. Sie können Dashboards zu einer Favoritenliste hinzufügen, über die Sie nicht nur auf Ihre favorisierten Dashboards, sondern auch auf Ihre kürzlich besuchten Dashboards zugreifen können. Weitere Informationen finden Sie unter Hinzufügen eines Dashboards zu Ihrer Favoritenliste.

Um auf CloudWatch Dashboards zugreifen zu können, benötigen Sie eine der folgenden Voraussetzungen:

- Die Richtlinie `AdministratorAccess`
- Die Richtlinie `CloudWatchFullAccess`

- Eine benutzerdefinierte Richtlinie mit einem oder mehreren dieser spezifischen Berechtigungen:
  - `cloudwatch:GetDashboard` und `cloudwatch:ListDashboards`, um Dashboards anzeigen zu können
  - `cloudwatch:PutDashboard`, um Dashboards erstellen oder ändern zu können
  - `cloudwatch:DeleteDashboards`, um Dashboards löschen zu können

Inhalt

Ein CloudWatch Dashboard erstellen

Erstellen Sie zunächst ein CloudWatch Dashboard. Sie können mehrere Dashboards erstellen und Dashboards zu einer Favoritenliste hinzufügen. Sie sind nicht auf die Anzahl der Dashboards beschränkt, die Sie in Ihrem AWS-Konto haben können. Alle Dashboards sind global. Sie sind nicht regionsspezifisch.

Das folgende Verfahren zeigt Ihnen, wie Sie ein Dashboard von der CloudWatch Konsole aus erstellen. Sie können die `PutDashboard`-API-Operation verwenden, um ein Dashboard über die Befehlszeilenschnittstelle zu erstellen. Die API-Operation enthält einen JSON-String, der den Inhalt Ihres Dashboards definiert. Weitere Informationen zum Erstellen eines Dashboards mit dem `PutDashboard` API-Vorgang finden Sie PutDashboardin der Amazon CloudWatch API-Referenz.

> **Tip**
>
> Wenn Sie mit der `PutDashboard`-API-Operation ein neues Dashboard erstellen, können Sie die JSON-Zeichenfolge aus einem bereits vorhandenen Dashboard verwenden.

So erstellen Sie ein Dashboard über die Konsole

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards, und wählen Sie dann Create Dashboard (Dashboard erstellen).
3. Geben Sie im Dialogfeld Create new dashboard (Neues Dashboard erstellen) einen Namen für das Dashboard ein, und wählen Sie anschließend Create dashboard (Dashboard erstellen).

   Wenn Sie den Namen CloudWatch-Default oder CloudWatch-Default- verwenden *`ResourceGroupName`*, erscheint das Dashboard in der Übersicht der CloudWatch Startseite unter Standard-Dashboard. Weitere Informationen finden Sie unter Erste Schritte mit Amazon CloudWatch.
4. Führen Sie im Dialogfeld Add to this dashboard (Zu diesem Dashboard hinzufügen) einen der folgenden Schritte aus:
   - Um dem Dashboard ein Diagramm hinzuzufügen, wählen Sie Line (Linie) oder Stacked area (Gestapelter Bereich) und dann Configure (Konfigurieren). Wählen Sie im Dialogfeld Add metric graph (Metrikdiagramm hinzufügen) die darzustellende(n) Metrik(en) aus, und wählen Sie dann Create widget (Widget erstellen). Wenn eine Metrik nicht im Dialogfeld erscheint, weil sie seit mehr als 14 Tagen keine Daten veröffentlicht hat, können Sie sie manuell hinzufügen. Weitere Informationen finden Sie unter Stellen Sie Metriken manuell auf einem CloudWatch Dashboard grafisch dar.
   - Um dem Dashboard eine Zahl hinzuzufügen, die eine Metrik anzeigt, wählen Sie Number (Zahl) und dann Configure (Konfigurieren). Wählen Sie im Dialogfeld Add metric graph (Metrikdiagramm hinzufügen) die darzustellende(n) Metrik(en) aus, und wählen Sie dann Create widget (Widget erstellen).
   - Um dem Dashboard einen Textblock hinzuzufügen, wählen Sie Text und dann Configure (Konfigurieren). Formatieren Sie im Dialogfeld New text widget (Neues Text-Widget) für Markdown Ihren Text mit Markdown, und wählen Sie dann Create widget (Widget erstellen).

5. (Optional) Wählen Sie Add widget (Widget hinzufügen) und wiederholen Sie dann Schritt 4, um dem Dashboard ein weiteres Widget hinzuzufügen. Sie können diesen Schritt beliebig oft wiederholen.

   Für jedes Diagramm auf dem Dashboard befindet sich oben rechts ein Informationssymbol. Wählen Sie dieses Symbol, um die Beschreibungen der Metriken im Diagramm zu sehen.

6. Wählen Sie Save dashboard aus.

CloudWatch Dashboard zur kontenübergreifenden Beobachtbarkeit

Wenn Sie mehrere AWS Konten haben, können Sie kontenübergreifende Observability einrichten und anschließend umfangreiche CloudWatch kontenübergreifende Dashboards in Ihren Monitoring-Konten erstellen. Sie können Ihre Metriken, Protokolle und Ablaufverfolgungen nahtlos und ohne Kontogrenzen suchen, visualisieren und analysieren.

Weitere Informationen zur Einrichtung CloudWatch kontenübergreifender Observability finden Sie unter. CloudWatch kontenübergreifende Beobachtbarkeit

Mit der CloudWatch kontoübergreifenden Beobachtbarkeit können Sie in einem Dashboard in einem Monitoring-Konto Folgendes tun:

- Suchen, betrachten und erstellen Sie Diagramme mit Metriken, die sich in Quellkonten befinden. Ein einzelnes Diagramm kann Metriken aus mehreren Konten enthalten.
- Erstellen Sie Alarme im Überwachungskonto, die Metriken in Quellkonten überwachen.
- Zeigen Sie die Protokollereignisse von Protokollgruppen an, die sich in Quellkonten befinden, und führen Sie CloudWatch Logs Insights-Abfragen von Protokollgruppen in Quellkonten durch. Eine einzelne CloudWatch Logs Insights-Abfrage in einem Überwachungskonto kann mehrere Protokollgruppen in mehreren Quellkonten gleichzeitig abfragen.
- Zeigen Sie Knoten von Quellkonten in einer Trace Map in X-Ray an. Anschließend können Sie die Karte nach bestimmten Quellkonten filtern.

Wenn Sie bei einem Monitoring-Konto angemeldet sind, erscheint oben rechts auf jeder Seite, die CloudWatch kontoübergreifende Observability-Funktionen unterstützt, ein blaues Monitoring-Konto-Logo.

Konto- und regionenübergreifende Dashboards

Sie können kontoübergreifende, regionsübergreifende Dashboards erstellen, die Ihre CloudWatch Daten aus mehreren AWS Konten und mehreren Regionen in einem Dashboard zusammenfassen. Von diesem übergeordneten Dashboard aus können Sie einen Überblick über Ihre gesamte Anwendung erhalten und auch auf spezifischere Dashboards aufgliedern, ohne sich von Konten anmelden und abmelden oder Regionen wechseln zu müssen.

Sie können kontoübergreifende, regionsübergreifende Dashboards im und programmgesteuert erstellen. AWS Management Console

Voraussetzung

Bevor Sie ein konto- und regionenübergreifendes Dashboard erstellen können, müssen Sie mindestens ein Freigabekonto und mindestens ein Überwachungskonto aktivieren. Um mit der Konsole ein kontoübergreifendes Dashboard erstellen zu können, müssen Sie die CloudWatch Konsole außerdem für kontoübergreifende Funktionen aktivieren. Weitere Informationen finden Sie unter Kontoübergreifende, regionsübergreifende Konsole CloudWatch .

Erstellen und Verwenden eines konto- und regionenübergreifenden Dashboards mit dem AWS Management Console

Sie können das verwenden, AWS Management Console um ein kontoübergreifendes regionsübergreifendes Dashboard zu erstellen.

So erstellen Sie ein konto- und regionenübergreifendes Dashboard

1. Melden Sie sich beim Überwachungskonto an.
2. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
3. Wählen Sie im Navigationsbereich Dashboards aus.
4. Wählen Sie ein Dashboard aus oder erstellen Sie ein neues Dashboard.
5. Sie können oben auf dem Bildschirm zwischen Konten und Regionen wechseln. Während Sie Ihr Dashboard erstellen, können Sie Widgets aus mehreren Konten und Regionen einbeziehen. Zu den Widgets gehören Grafiken, Alarme und CloudWatch Logs Insights-Widgets.

Erstellen eines Diagramms mit Metriken aus verschiedenen Konten und Regionen

1. Melden Sie sich beim Überwachungskonto an.

2. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
3. Wählen Sie im Navigationsbereich Metrics (Metriken) und dann All metrics (Alle Metriken) aus.
4. Wählen Sie das Konto und die Region aus, aus der Sie Metriken hinzufügen möchten. Sie können Ihr Konto und Ihre Region in den Dropdown-Menüs für das Konto und die Region oben rechts auf dem Bildschirm auswählen.
5. Fügen Sie dem Diagramm die gewünschten Metriken hinzu. Weitere Informationen finden Sie unter Grafisches Darstellen von Metriken.
6. Wiederholen Sie die Schritte 4-5, um Metriken aus anderen Konten und Regionen hinzuzufügen.
7. (Optional) Wählen Sie die Registerkarte Graphed metrics (Metriken mit Diagrammen) und fügen Sie eine Metrik-Mathematik-Funktion hinzu, die die ausgewählten Metriken verwendet. Weitere Informationen finden Sie unter Verwenden von Metrikberechnungen.

   Sie können auch ein einzelnes Diagramm so einrichten, dass es mehrere `SEARCH`-Funktionen enthält. Jede Suche kann sich auf ein anderes Konto oder eine andere Region beziehen.
8. Wenn Sie mit dem Diagramm fertig sind, wählen Sie Actions (Aktionen), Add to Dashboard (Zum Dashboard hinzufügen).

   Wählen Sie Ihr kontoübergreifendes Dashboard aus und wählen Sie Add to dashboard (Zu Dashboard hinzufügen).

Hinzufügen eines Alarms von einem anderen Konto zu Ihrem kontoübergreifenden Dashboard

1. Melden Sie sich beim Überwachungskonto an.
2. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
3. Wählen Sie oben auf der Seite das Konto aus, in dem sich der Alarm befindet.
4. Klicken Sie im Navigationsbereich auf Alarms (Alarme).
5. Aktivieren Sie das Kontrollkästchen neben dem Alarm, den Sie hinzufügen möchten, und wählen Sie Add to dashboard (Zum Dashboard hinzufügen).
6. Wählen Sie das kontoübergreifende Dashboard aus, dem Sie es hinzufügen möchten, und wählen Sie Add to dashboard (Zu Dashboard hinzufügen).

Programmgesteuertes Erstellen eines konto- und regionenübergreifenden Dashboards

Sie können die AWS APIs und SDKs verwenden, um Dashboards programmgesteuert zu erstellen. Weitere Informationen finden Sie unter PutDashboard

Um konto- und regionenübergreifende Dashboards zu aktivieren, haben wir der Dashboard-Body-Struktur neue Parameter hinzugefügt, wie in der folgenden Tabelle und den Beispielen gezeigt. Weitere Informationen zur gesamten Dashboard-Body-Struktur finden Sie unter Dashboard-Body-Struktur und Syntax.

Parameter	Verwenden Sie	Scope	Standard
`accountId`	Gibt die ID des Kontos an, in dem sich das Widget oder die Metrik befindet.	Widget oder Metrik	Konto, das derzeit angemeldet ist
`region`	Gibt die Region der Metrik an.	Widget oder Metrik	Aktuell in der Konsole ausgewählte Region

Die folgenden Beispiele veranschaulichen die JSON-Quelle für Widgets in einem konto- und regionenübergreifenden Dashboard.

In diesem Beispiel wird das `accountId`-Feld auf die ID des Freigabekontos auf Widgetebene festgelegt. Dies gibt an, dass alle Metriken in diesem Widget von diesem Freigabekonto und dieser Region stammen.

```
{
    "widgets": [
        {
            ...
            "properties": {
              "metrics": [
                     …..
                     ],
                    "accountId": "111122223333",
                    "region": "us-east-1"
          }
```

```
        }
    ]
}
```

In diesem Beispiel wird das `accountId`-Feld auf der Ebene jeder Metrik unterschiedlich festgelegt. In diesem Beispiel stammen die verschiedenen Metriken in diesem metrischen mathematischen Ausdruck von verschiedenen Freigabekonten und verschiedenen Regionen.

```
{
"widgets": [
    {
        ...
        "properties": {
            "metrics": [
                [ { "expression": "SUM(METRICS())", "label": "[avg: ${AVG}]
 Expression1", "id": "e1", "stat": "Sum" } ],
                [ "AWS/EC2", "CPUUtilization", { "id": "m2", "accountId":
 "5555666677778888", "region": "us-east-1", "label": "[avg: ${AVG}] ApplicationALabel
 " } ],
                [ ".", ".", { "id": "m1", "accountId": "9999000011112222", "region":
 "eu-west-1", "label": "[avg: ${AVG}] ApplicationBLabel" } ]
            ],
            "view": "timeSeries",
            "region":"us-east-1", ---> home region of the metric. Not present in above
 example
            "stacked": false,
            "stat": "Sum",
            "period": 300,
            "title": "Cross account example"
        }
    }
]
}
```

Dieses Beispiel zeigt ein Alarm-Widget.

```
{
    "type": "metric",
    "x": 6,
    "y": 0,
    "width": 6,
    "height": 6,
    "properties": {
```

```
        "accountID": "111122223333",
        "title": "over50",
        "annotations": {
            "alarms": [
                "arn:aws:cloudwatch:us-east-1:379642911888:alarm:over50"
            ]
        },
        "view": "timeSeries",
        "stacked": false
    }
}
```

Dieses Beispiel bezieht sich auf ein CloudWatch Logs Insights-Widget.

```
{
    "type": "log",
    "x": 0,
    "y": 6,
    "width": 24,
    "height": 6,
    "properties": {
        "query": "SOURCE 'route53test' | fields @timestamp, @message\n| sort @timestamp
 desc\n| limit 20",
        "accountId": "111122223333",
        "region": "us-east-1",
        "stacked": false,
        "view": "table"
    }
}
```

Eine andere Möglichkeit, Dashboards programmgesteuert zu erstellen, besteht darin, zuerst eines im Dashboard zu erstellen und dann die AWS Management Console JSON-Quelle dieses Dashboards zu kopieren. Laden Sie dazu das Dashboard und wählen Sie Actions (Aktionen), View/edit source (Quelle anzeigen/bearbeiten). Anschließend können Sie dieses Dashboard-JSON kopieren, um es als Vorlage zum Erstellen ähnlicher Dashboards zu verwenden.

Flexible Dashboards mit Dashboard-Variablen erstellen

Verwenden Sie Dashboard-Variablen, um flexible Dashboards zu erstellen, die je nach dem Wert eines Eingabefeldes innerhalb des Dashboards schnell unterschiedliche Inhalte in mehreren Widgets anzeigen können. Sie können beispielsweise ein Dashboard erstellen, das schnell zwischen

verschiedenen Lambda-Funktionen oder Amazon EC2 EC2-Instance-IDs wechseln kann, oder ein Dashboard, das zu verschiedenen AWS Regionen wechseln kann.

Nachdem Sie ein Dashboard erstellt haben, das eine Variable verwendet, können Sie dasselbe Variablenmuster in andere bestehende Dashboards kopieren.

Die Verwendung von Dashboard-Variablen verbessert die Arbeitsabläufe der Personen, die Ihre Dashboards verwenden. Es kann auch Ihre Kosten senken, da Sie Dashboard-Variablen in einem einzigen Dashboard verwenden, anstatt mehrere ähnliche Dashboards zu erstellen.

> **Note**
>
> Wenn Sie ein Dashboard freigeben, das Dashboard-Variablen enthält, können die Personen, für die Sie es freigeben, nicht zwischen den Variablenwerten wechseln.

Typen von Dashboard-Variablen

Die Dashboard-Variable kann eine Eigenschaftsvariable oder eine Mustervariable sein.

- Eigenschaftsvariablen ändern alle Instances einer Eigenschaft in allen Widgets im Dashboard. Bei dieser Eigenschaft kann es sich um eine beliebige JSON-Eigenschaft in der JSON-Quelle eines Dashboards handeln, z. B. `region`. Es kann sich auch um einen Dimensionsnamen für eine Metrik handeln, z. B. `InstanceID` oder `FunctionName`.

  Ein Tutorial, das eine Eigenschaftsvariable verwendet, finden Sie unter Tutorial: Ein Lambda-Dashboard mit dem Funktionsnamen als Variable erstellen.

  Weitere Informationen über die JSON-Quelle von Dashboards finden Sie unter Dashboard Body Structure and Syntax. In der CloudWatch Konsole können Sie die JSON-Quelle für jedes benutzerdefinierte Dashboard sehen, indem Sie Aktionen, Quelle anzeigen/bearbeiten auswählen.
- Mustervariablen verwenden ein Muster eines regulären Ausdrucks, um eine JSON-Eigenschaft vollständig oder nur in einem bestimmten Teil zu ändern.

  Ein Tutorial, das eine Mustervariable verwendet, finden Sie unter Tutorial: Ein Dashboard erstellen, das ein Muster mit regulären Ausdrücken verwendet, um zwischen Regionen zu wechseln.

Eigenschaftsvariablen eignen sich für die meisten Anwendungsfälle und sind weniger komplex in der Einrichtung.

Themen

Tutorial: Ein Lambda-Dashboard mit dem Funktionsnamen als Variable erstellen

Die Schritte in diesem Verfahren veranschaulichen, wie Sie mithilfe einer Eigenschaftsvariablen ein flexibles Dashboard erstellen, das eine Vielzahl von Metrikdiagrammen anzeigt. Dazu gehört ein Drop-down-Auswahlfeld auf dem Dashboard, mit dem Sie die Metriken in allen Diagrammen zwischen verschiedenen Lambda-Funktionen umschalten können.

Weitere Anwendungsbeispiele für diese Art von Dashboard sind die Verwendung von `InstanceId` als Variable zur Erstellung eines Dashboards mit Metriken und einem Dropdown-Menü für Instance-IDs. Alternativ könnten Sie ein Dashboard erstellen, das `region` als Variable verwendet, um dieselben Metriken aus verschiedenen Regionen anzuzeigen.

So verwenden Sie eine Dashboard-Eigenschaftsvariable, um ein flexibles Lambda-Dashboard zu erstellen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Dashboards und Create dashboard aus.
3. Geben Sie einen Namen für das Dashboard ein und wählen Sie Dashboard erstellen.
4. Fügen Sie dem Dashboard Widgets hinzu, die Metriken für eine Lambda-Funktion anzeigen. Wenn Sie diese Widgets erstellen, geben Sie Lambda, Nach Funktionsname für die Widget-Metriken an. Für die Funktion geben Sie eine der Lambda-Funktionen an, die Sie in dieses Dashboard aufnehmen möchten.

   Weitere Informationen zum Hinzufügen von Widgets zu einem Dashboard finden Sie unter Widgets auf CloudWatch Dashboards erstellen und damit arbeiten.
5. Nachdem Sie die Widgets hinzugefügt haben, wählen Sie während der Anzeige des Dashboards Aktionen, Variablen, Variable erstellen aus.
6. Wählen Sie Eigenschaftsvariable aus.
7. Wählen Sie für Eigenschaft, die sich durch die Variable ändert, die Option FunctionName.

8. Für den Eingabetyp empfehlen wir für diesen Anwendungsfall die Wahl von Auswahlmenü (Dropdown). Dadurch wird im Dashboard ein Dropdownmenü erstellt, in dem Sie den Namen der Lambda-Funktion auswählen können, für die Metriken angezeigt werden sollen.

   Wenn dies für ein Dashboard ist, das nur zwischen zwei oder drei verschiedenen Werten für eine Variable hin- und herschaltet, wäre Optionsfeld eine gute Wahl.

   Wenn Sie lieber Werte für die Variable eingeben oder einfügen möchten, wählen Sie Texteingabe. Diese Option enthält weder eine Dropdownliste noch Optionsfelder.
9. Wenn Sie Auswahlmenü (Dropdown) wählen, müssen Sie dann entscheiden, ob Sie das Menü durch Eingabe von Werten oder mithilfe einer Metriksuche füllen möchten. Gehen wir für diesen Anwendungsfall davon aus, dass Sie über eine große Anzahl von Lambda-Funktionen verfügen und nicht alle manuell eingeben möchten. Wählen Sie Ergebnisse einer Metriksuche verwenden aus und gehen Sie dann wie folgt vor:

   a. Wählen Sie Vordefinierte Abfragen, Lambda, Fehler aus.

      (Wenn Sie Fehler auswählen, wird die Metrik Fehler nicht zum Dashboard hinzugefügt. Es füllt das FunctionNameVariablenauswahlfeld jedoch schnell aus.)

   b. Wählen Sie Nach Funktionsname und anschließend Suchen.

      Unter der Schaltfläche Suchen wird dann die FunctionNameOption ausgewählt angezeigt. Außerdem wird eine Meldung darüber angezeigt, wie viele FunctionNameDimensionswerte gefunden wurden, um das Eingabefeld auszufüllen.
10. (Optional) Wählen Sie für weitere Einstellungen die Option Sekundäre Einstellungen und führen Sie einen oder mehrere der folgenden Schritte aus:

    - Um den Namen Ihrer Variablen anzupassen, geben Sie den Namen im Feld Benutzerdefinierter Variablenname ein.
    - Um die Bezeichnung für das Variableneingabefeld anzupassen, geben Sie die Bezeichnung im Feld Eingabebezeichnung ein.
    - Um den Standardwert für diese Variable festzulegen, wenn das Dashboard zum ersten Mal geöffnet wird, geben Sie den Standardwert in das Feld Standardwert ein.
11. Wählen Sie Variable hinzufügen aus.

    Ein FunctionNameDrop-down-Auswahlfeld wird oben im Dashboard angezeigt. Sie können in diesem Feld eine Lambda-Funktion auswählen, und alle Widgets, die die Variable verwenden, zeigen Informationen über die ausgewählte Funktion an.

Wenn Sie dem Dashboard später weitere Widgets hinzufügen, die Lambda-Metriken mit der FunctionNameDimension überwachen, verwenden diese automatisch die Variable.

Tutorial: Ein Dashboard erstellen, das ein Muster mit regulären Ausdrücken verwendet, um zwischen Regionen zu wechseln

Die Schritte in diesem Verfahren veranschaulichen, wie Sie ein flexibles Dashboard erstellen, das zwischen Regionen wechseln kann. In diesem Tutorial wird anstelle einer Eigenschaftsvariablen eine Mustervariable mit regulären Ausdrücken verwendet. Ein Tutorial, das eine Eigenschaftsvariable verwendet, finden Sie unter Tutorial: Ein Lambda-Dashboard mit dem Funktionsnamen als Variable erstellen.

Für viele Anwendungsfälle können Sie ein Dashboard erstellen, das mithilfe einer Eigenschaftsvariablen zwischen Regionen wechselt. Wenn die Widgets jedoch auf Amazon-Ressourcennamen (ARNs) basieren, die Regionsnamen enthalten, müssen Sie eine Mustervariable verwenden, um die Regionsnamen innerhalb der ARNs zu ändern.

So verwenden Sie eine Dashboard-Mustervariable, um ein flexibles Dashboard mit mehreren Regionen zu erstellen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Dashboards und Create dashboard aus.
3. Geben Sie einen Namen für das Dashboard ein und wählen Sie Dashboard erstellen.
4. Fügen Sie dem Dashboard Widgets hinzu. Wenn Sie die Widgets hinzufügen, für die Sie regionsspezifische Daten anzeigen möchten, vermeiden Sie es, Dimensionen mit Werten anzugeben, die nur in einer Region vorkommen. Geben Sie beispielsweise für Amazon-EC2-Metriken aggregierte Metriken an, anstatt Metriken, die InstanceID als Dimension verwenden.

   Weitere Informationen zum Hinzufügen von Widgets zu einem Dashboard finden Sie unter Widgets auf CloudWatch Dashboards erstellen und damit arbeiten.
5. Nachdem Sie die Widgets hinzugefügt haben, wählen Sie während der Anzeige des Dashboards Aktionen, Variablen, Variable erstellen aus.
6. Wählen Sie Mustervariable aus.
7. Geben Sie unter Eigenschaft, die sich durch die Variable ändert, den Namen der aktuellen Dashboard-Region ein, z. B. **`us-east-2`**.

Wenn die Bezeichnung unter dem Feld die Widgets anzeigt, auf die sich die Variable auswirkt, haben Sie die richtige Region eingegeben.

8. Als Eingabetyp wählen Sie für diesen Anwendungsfall Optionsfeld.
9. Wählen Sie unter Definieren, wie Eingaben gefüllt werden, die Option Liste mit benutzerdefinierten Werten erstellen aus.
10. Geben Sie unter Benutzerdefinierte Werte erstellen die Regionen ein, zwischen denen Sie wechseln möchten, mit einer Region in jeder Zeile. Geben Sie nach jeder Region ein Komma und dann die Bezeichnung ein, die für dieses Optionsfeld angezeigt werden soll. Beispielsweise:

    ```
    us-east-1, N. Virginia

    us-east-2, Ohio

    eu-west-3, Paris
    ```

    Wenn Sie die benutzerdefinierten Werte eingeben, wird das Vorschaufenster aktualisiert und zeigt an, wie die Optionsfelder aussehen werden.
11. (Optional) Wählen Sie für weitere Einstellungen die Option Sekundäre Einstellungen und führen Sie einen oder mehrere der folgenden Schritte aus:

    - Um den Namen Ihrer Variablen anzupassen, geben Sie den Namen im Feld Benutzerdefinierter Variablenname ein.
    - Um die Bezeichnung für das Variableneingabefeld anzupassen, geben Sie die Bezeichnung im Feld Eingabebezeichnung ein. Geben Sie für dieses Tutorial **`Region:`** ein.

      Wenn Sie hier einen Wert eingeben, wird das Vorschaufenster aktualisiert und zeigt an, wie die Optionsfelder aussehen werden.
    - Um den Standardwert für diese Variable festzulegen, wenn das Dashboard zum ersten Mal geöffnet wird, geben Sie den Standardwert in das Feld Standardwert ein.
12. Wählen Sie Variable hinzufügen aus.

    Das Dashboard wird angezeigt, mit der Bezeichnung Region: neben den Optionsfeldern für die Regionen am oberen Rand. Wenn Sie zwischen den Regionen wechseln, zeigen alle Widgets, die die Variable verwenden, Informationen über die ausgewählte Region an.

Eine Variable in ein anderes Dashboard kopieren

Nachdem Sie ein Dashboard mit nützlichen Variablen erstellt haben, können Sie diese Variablen in andere vorhandene Dashboards kopieren. Weitere Informationen zu Dashboard-Variablen finden Sie unter Flexible Dashboards mit Dashboard-Variablen erstellen.

Um eine Dashboard-Variable in ein anderes Dashboard zu kopieren

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Dashboards und dann den Namen des Dashboards, das die Variable enthält, die Sie kopieren möchten. Geben Sie bei Bedarf eine Zeichenfolge ein, um Dashboards mit passenden Namen zu finden.
3. Wählen Sie Aktionen, Variablen, Variablen verwalten aus.
4. Wählen Sie das Optionsfeld neben der Variablen, die Sie kopieren möchten, und wählen Sie In anderes Dashboard kopieren aus.
5. Wählen Sie das Auswahlfeld und beginnen Sie mit der Eingabe des Dashboard-Namens, in den Sie die Variable kopieren möchten.
6. Wählen Sie den Namen des Dashboards aus und wählen Sie Variable kopieren.

Widgets auf CloudWatch Dashboards erstellen und damit arbeiten

Verwenden Sie die Themen in diesem Abschnitt, um Diagramme, Alarme und Text-Widgets in Ihren Dashboards zu erstellen und zu bearbeiten.

Inhalt

- Fügen Sie ein Text-Widget zu einem CloudWatch Dashboard hinzu oder entfernen Sie es
- Fügen Sie ein Alarm-Widget zu einem CloudWatch Dashboard hinzu oder entfernen Sie es
- Fügen Sie ein Datentabellen-Widget zu einem CloudWatch Dashboard hinzu oder entfernen Sie es
- Diagramme auf einem CloudWatch Dashboard verknüpfen und deren Verknüpfung aufheben

Ein Diagramm zu einem CloudWatch Dashboard hinzufügen oder daraus entfernen

Sie können Ihrem CloudWatch Dashboard Grafiken hinzufügen, die eine oder mehrere Metriken enthalten. Zu den Arten von Diagrammen, die Sie Ihrem Dashboard hinzufügen können, gehören Line (Linie), Stacked Area (Gestapelter Bereich), Number (Zahl), Gauge (Messinstrument), Bar (Balken) und Pie (Kreis). Sie können Diagramme aus Ihrem Dashboard entfernen, wenn Sie sie nicht mehr benötigen. In diesem Abschnitt wird beschrieben, wie Sie Diagramme zu Ihrem Dashboard hinzufügen oder daraus entfernen können. Informationen zum Bearbeiten eines Diagramms in Ihrem Dashboard finden Sie unter Bearbeiten eines Diagramms in einem CloudWatch Dashboard.

So fügen Sie einem Dashboard ein Diagramm hinzu

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie das Symbol + und dann den Diagrammtyp, den Sie zu Ihrem Dashboard hinzufügen möchten, und wählen Sie dann Weiter.

   - Wenn Sie die Option Line (Linie), Stacked area (Gestapelter Bereich), Bar (Balken), oder Pie (Kreis), wählen Sie Metrics (Metriken).
4. Suchen Sie auf der Registerkarte Durchsuchen nach den Metriken, die Sie grafisch darstellen möchten, und wählen Sie die gewünschten Metriken aus.
5. (Optional) Um den Zeitbereich Ihres Diagramms zu ändern, wählen Sie einen der vordefinierten Zeitbereiche im oberen Teil des Bildschirms. Die Zeitspanne reicht von 1 Stunde bis 1 Woche (1h, 3h, 12h, 1d, 3d, oder 1w).

   Um Ihren eigenen Zeitraum festzulegen, wählen Sie Custom (Benutzerdefiniert).

   - (Optional) Damit dieses Widget weiterhin den von Ihnen ausgewählten Zeitraum verwendet, auch wenn der Zeitraum für den Rest des Dashboards später geändert wird, wählen Sie Zeitraum beibehalten.

6. (Optional) Um den Widget-Typ Ihres Diagramms zu ändern, verwenden Sie das Dropdown-Menü neben den vordefinierten Zeitbereichen.
7. (Optional) In Graphed metrics (Grafisch dargestellte Metriken) können Sie Ihrer Metrik eine dynamische Bezeichnung hinzufügen und die Bezeichnung, die Farbe der Bezeichnung, die Statistik und den Zeitraum Ihrer Metrik ändern. Sie können auch die Position der Beschriftungen auf der Y-Achse von links nach rechts festlegen.

   a. Um eine dynamische Beschriftung hinzuzufügen, wählen Sie Graphed metrics (Grafisch dargestellte Metriken) und Add dynamic labels (Dynamische Beschriftungen hinzufügen). Dynamische Beschriftungen zeigen in der Diagrammlegende eine Statistik zu Ihrer Metrik an. Dynamische Labels werden automatisch aktualisiert, wenn Ihr Dashboard oder Diagramm aktualisiert wird. Standardmäßig werden die dynamischen Werte, die Sie den Etiketten hinzufügen, am Anfang der Etiketten angezeigt. Weitere Informationen finden Sie unter Dynamische Labels verwenden.

   b. Um die Farbe einer Metrik zu ändern, wählen Sie das Farbquadrat neben der Metrik.

   c. Um die Statistik zu ändern, wählen Sie das Dropdown-Menü unter Statistic (Statistik) und dann einen neuen Wert aus. Weitere Informationen finden Sie unter Statistics (Statistiken).

   d. Um den Zeitraum zu ändern, wählen Sie das Dropdown-Menü unter der Spalte Period (Zeitraum) und wählen Sie dann einen neuen Wert.
8. Wenn Sie ein Messinstrument-Widget erstellen, müssen Sie die Registerkarte Optionen wählen und die Min- und Max-Werte für die beiden Endpunkte des Messgerätes angeben.
9. (Optional) Um die Y-Achse anzupassen, wählen Sie Options (Optionen). Sie können unter der Left Y-axis (Linke Y-Achse) im Beschriftungsfeld eine benutzerdefinierte Beschriftung hinzufügen. Wenn Ihr Diagramm Werte auf der rechten Seite der Y-Achse anzeigt, können Sie auch diese Beschriftung anpassen. Sie können auch Minimal- und Maximalwerte für die Y-Achse festlegen, so dass Ihr Diagramm nur die von Ihnen angegebenen Wertebereiche anzeigt.
10. (Optional) Um horizontale Anmerkungen zu Linien- oder gestapelten Flächendiagrammen hinzuzufügen oder zu bearbeiten, oder um Schwellenwerte zu Messinstrument-Widgets hinzuzufügen, wählen Sie Optionen:

    a. Um eine horizontale Anmerkung oder einen Schwellenwert hinzuzufügen, wählen Sie Horizontale Anmerkung hinzufügen oder Schwellenwert hinzufügen.

    b. Geben Sie unter Bezeichnung eine Bezeichnung für die Anmerkung ein und wählen Sie dann das Häkchensymbol.

c. Wählen Sie für Value (Wert) das Stift- und Papiersymbol neben dem aktuellen Wert und geben Sie Ihren neuen Wert ein. Wählen Sie nach der Eingabe Ihres Wertes das Häkchen aus.

d. Wählen Sie für Fill (Füllung) das Dropdown-Menü und geben Sie an, wie die Schattierung in Ihrer Anmerkung verwendet werden soll. Sie können None (Keine), Above (Über), Between (Zwischen) oder Below (Unter) auswählen. Um die Farbe der Füllung zu ändern, wählen Sie das Farbquadrat, das sich neben der Anmerkung befindet.

e. Legen Sie für Axis (Achse) fest, ob Ihre Anmerkung auf der linken oder rechten Seite der Y-Achse angezeigt wird.

f. Um eine Anmerkung auszublenden, deaktivieren Sie das Kontrollkästchen neben der Anmerkung, die Sie ausblenden möchten.

g. Um eine Anmerkung zu löschen, wählen Sie X unter Actions (Aktionen).

> **Note**
>
> Sie können diese Schritte wiederholen, um mehrere horizontale Anmerkungen oder Schwellenwerte zum selben Diagramm oder Messgerät hinzuzufügen.

11. (Optional) Um vertikale Anmerkungen hinzuzufügen oder zu bearbeiten, wählen Sie Options (Optionen):

    a. Wenn Sie eine vertikale Anmerkung hinzufügen möchten, klicken Sie auf Add vertical annotation (Vertikale Anmerkung hinzufügen).

    b. Wählen Sie für Label (Bezeichnung) das Stift- und Papiersymbol neben der aktuellen Anmerkung und geben Sie Ihre neue Anmerkung ein. Wenn Sie nur das Datum und die Uhrzeit anzeigen möchten, lassen Sie das Beschriftungsfeld leer.

    c. Wählen Sie unter Date (Datum) das aktuelle Datum und die aktuelle Uhrzeit und geben Sie das neue Datum und die neue Uhrzeit ein.

    d. Wählen Sie für Fill (Füllung) die Dropdown-Liste aus, und geben Sie an, wie die Schattierung in Ihrer Anmerkung verwendet werden soll. Sie können None (Keine), Above (Über), Between (Zwischen) oder Below (Unter) auswählen. Um die Farbe der Füllung zu ändern, wählen Sie das Farbquadrat, das sich neben der Anmerkung befindet.

    e. Um eine Anmerkung auszublenden, deaktivieren Sie das Kontrollkästchen neben der Anmerkung, die Sie ausblenden möchten.

    f. Um eine Anmerkung zu löschen, wählen Sie X unter Actions (Aktionen).

> **Note**
>
> Sie können die Schritte wiederholen um einem Diagramm mehrere vertikale Anmerkungen hinzuzufügen.

12. Wählen Sie Create widget aus.
13. Wählen Sie Save dashboard aus.

So entfernen Sie ein Diagramm von einem Dashboard

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie in der oberen rechten Ecke des Diagramms, das Sie entfernen möchten, die Option Widget actions (Widget-Aktionen) und anschließend die Option Delete (Löschen).
4. Wählen Sie Save dashboard aus.

Stellen Sie Metriken manuell auf einem CloudWatch Dashboard grafisch dar

Wenn eine Metrik in den letzten 14 Tagen keine Daten veröffentlicht hat, können Sie sie nicht finden, wenn Sie nach Metriken suchen, die Sie einem Diagramm auf einem CloudWatch Dashboard hinzufügen möchten. Gehen Sie wie folgt vor, um einem vorhandenen Diagramm manuell eine Metrik hinzuzufügen.

So fügen Sie eine Metrik, die Sie bei der Suche nicht finden können, zu einem Diagramm hinzu:

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Das Dashboard muss bereits ein Diagramm enthalten, dem Sie die Metrik hinzufügen können. Wenn dies nicht der Fall ist, erstellen Sie das Diagramm und fügen Sie ihm eine beliebige Metrik hinzu. Weitere Informationen finden Sie unter Ein Diagramm zu einem CloudWatch Dashboard hinzufügen oder daraus entfernen .
4. Wählen Sie Actions (Aktionen), View/edit source (Quelle anzeigen/bearbeiten).

   Ein JSON-Block wird angezeigt. Der Block gibt die Widgets auf dem Dashboard und ihren Inhalt an. Das folgende Beispiel zeigt einen Teil dieses Blocks, der ein Diagramm definiert.

```
{
            "type": "metric",
            "x": 0,
            "y": 0,
            "width": 6,
            "height": 3,
            "properties": {
                "view": "singleValue",
                "metrics": [
                    [ "AWS/EBS", "VolumeReadOps", "VolumeId",
 "vol-1234567890abcdef0" ]
                ],
                "region": "us-west-1"
            }
        },
```

In diesem Beispiel definiert der nachfolgende Abschnitt die in diesem Diagramm angezeigte Metrik.

```
[ "AWS/EBS", "VolumeReadOps", "VolumeId", "vol-1234567890abcdef0" ]
```

5. Fügen Sie ein Komma nach der schließenden Klammer hinzu, falls noch nicht vorhanden, und fügen Sie dann einen ähnlichen Klammerabschnitt nach dem Komma hinzu. Legen Sie in diesem neuen Abschnitt den Namespace, den Metrik-Namen und alle erforderlichen Dimensionen der Metrik fest, die Sie dem Diagramm hinzufügen. Im Folgenden wird ein Beispiel gezeigt.

```
[ "AWS/EBS", "VolumeReadOps", "VolumeId", "vol-1234567890abcdef0" ],
[ "MyNamespace", "MyMetricName", "DimensionName", "DimensionValue" ]
```

Weitere Informationen zur Formatierung von Metriken in JSON finden Sie unter Eigenschaften eines Metrik-Widget-Objekts.

6. Wählen Sie Aktualisieren.

Arbeiten mit bestehenden Diagrammen

Befolgen Sie die Verfahren in diesen Abschnitten, um Ihre vorhandenen Dashboard-Diagramm-Widgets zu bearbeiten und zu ändern.

Themen

Bearbeiten Sie ein Diagramm auf einem Dashboard CloudWatch

Sie können die Grafiken bearbeiten, die Sie Ihrem CloudWatch Dashboard hinzufügen. Sie können den Titel, die Statistik oder den Zeitraum eines Diagramms ändern. Sie können Metriken zu Ihren Diagrammen hinzufügen, aktualisieren und entfernen. Wenn Ihr Diagramm mehr als eine Metrik enthält, können Sie die nicht verwendeten Metriken ausblenden, um die Übersichtlichkeit zu erhöhen. In diesem Abschnitt wird beschrieben, wie Sie ein Diagramm in Ihrem Dashboard bearbeiten können. Informationen zum Erstellen eines Diagramms finden Sie unter Hinzufügen oder Entfernen eines Diagramms aus einem CloudWatch Dashboard.

New interface

So bearbeiten Sie ein Diagramm auf einem Dashboard

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie in der oberen rechten Ecke des Diagramms, das Sie bearbeiten möchten, die Option Widget actions (Widget-Aktionen) und anschließend die Option Edit (Bearbeiten).
4. Um den Titel des Diagramms zu ändern, wählen Sie das Stift- und Papiersymbol, das sich neben dem aktuellen Titel befindet. Geben Sie den neuen Titel ein und wählen Sie dann Apply (Anwenden) aus.
5. (Optional) Um den Zeitbereich Ihres Diagramms zu ändern, wählen Sie einen der vordefinierten Zeitbereiche im oberen Bereich des Diagramms. Die Zeitspanne reicht von 1 Stunde bis 1 Woche (1h, 3h, 12h, 1d, 3d, oder 1w).

   Um Ihren eigenen Zeitraum festzulegen, wählen Sie Custom (Benutzerdefiniert).

   - (Optional) Damit dieses Widget weiterhin den von Ihnen ausgewählten Zeitraum verwendet, auch wenn der Zeitraum für den Rest des Dashboards später geändert wird, wählen Sie Zeitraum beibehalten.

6. Um den Widget-Typ Ihres Diagramms zu ändern, verwenden Sie das Dropdown-Menü, das sich neben den vordefinierten Zeitbereichen befindet.
7. In Graphed metrics (Grafisch dargestellte Metriken) können Sie Ihrer Metrik eine dynamische Bezeichnung hinzufügen und die Bezeichnung, die Farbe der Bezeichnung, die Statistik und den Zeitraum der Metrik ändern. Sie können auch die Position der Beschriftungen auf der Y-Achse von links nach rechts festlegen.

   a. Um eine dynamische Bezeichnung für eine Metrik hinzuzufügen, wählen Sie Dynamic labels (Dynamische Labels). Dynamische Labels zeigen eine Statistik über die Metrik in der Diagrammlegende an. Dynamische Labels werden automatisch aktualisiert, wenn Ihr Dashboard oder Diagramm aktualisiert wird. Standardmäßig erscheinen die dynamischen Werte, die Sie den Etiketten hinzufügen, am Anfang der Etiketten. Weitere Informationen finden Sie unter Dynamische Labels verwenden.

   b. Um die Farbe einer Metrik zu ändern, wählen Sie das Farbquadrat neben der Metrik.

   c. Um die Statistik zu ändern, wählen Sie den statistischen Wert in der Spalte Statistic (Statistik) und dann einen neuen Wert aus. Weitere Informationen finden Sie unter Statistiken.

   d. Um die Periode zu ändern, wählen Sie den Periodenwert in der Spalte Period (Periode) und dann einen neuen Wert aus.
8. Um horizontale Anmerkungen hinzuzufügen oder zu bearbeiten, wählen Sie Options (Optionen) aus:

   a. Wenn Sie eine horizontale Anmerkung hinzufügen möchten, wählen Sie Add horizontal annotation aus.

   b. Wählen Sie für Label (Markierung) das Stift-und-Papier-Symbol neben der aktuellen Anmerkung aus. Geben Sie dann Ihre neue Anmerkung ein. Wählen Sie das Häkchen aus, nachdem Sie Ihre Anmerkung eingegeben haben.

   c. Wählen Sie für Value (Wert) das Stift-und-Papiersymbol neben dem aktuellen Metrikwert aus. Geben Sie dann Ihren neuen Metrikwert ein. Wählen Sie nach der Eingabe Ihres Wertes das Häkchen aus.

   d. Wählen Sie für Fill (Füllen) das Dropdown-Menü unter der Spalte aus und legen Sie dann fest, wie Ihre Anmerkung die Schattierung verwenden soll. Sie können None (Keine), Above (Über), Between (Zwischen) oder Below (Unter) auswählen. Wenn Sie Between (Zischen) auswählen, wird ein weiteres neues Markierungs- und Wertefeld angezeigt.

> **Tip**
>
> Sie können die Füllfarbe ändern, indem Sie das farbige Quadrat neben der Anmerkung auswählen.

e. Legen Sie für Axis (Achse) fest, ob Ihre Anmerkung auf der linken oder rechten Seite der Y-Achse angezeigt wird.

f. Um eine Anmerkung auszublenden, deaktivieren Sie das Kontrollkästchen neben der Anmerkung, die Sie im Diagramm ausblenden möchten.

g. Um eine Anmerkung zu löschen, wählen Sie das X in der Spalte Actions (Aktionen) aus.

> **Note**
>
> Sie können die Schritte wiederholen, um einem Diagramm mehrere horizontale Anmerkungen hinzuzufügen.

9. Um vertikale Anmerkungen hinzuzufügen oder zu bearbeiten, wählen Sie Options (Optionen) aus:

   a. Wenn Sie eine vertikale Anmerkung hinzufügen möchten, klicken Sie auf Add vertical annotation (Vertikale Anmerkung hinzufügen).

   b. Wählen Sie für Label (Markierung) das Stift-und-Papier-Symbol neben der aktuellen Anmerkung aus. Geben Sie dann Ihre neue Anmerkung ein. Wählen Sie das Häkchen aus, nachdem Sie Ihre Anmerkung eingegeben haben.

   > **Tip**
   >
   > Um nur Datum und Uhrzeit anzuzeigen, lassen Sie das Markierungsfeld leer.

   c. Wählen Sie für Date (Datum) das aktuelle Datum und die Uhrzeit aus. Geben Sie dann das neue Datum und die neue Uhrzeit ein.

   d. Wählen Sie für Fill (Füllen) das Dropdown-Menü unter der Spalte aus und legen Sie dann fest, wie Ihre Anmerkung die Schattierung verwenden soll. Sie können None (Keine), Above (Über), Between (Zwischen) oder Below (Unter) auswählen. Wenn Sie Between (Zwischen) auswählen, wird ein neues Markierungs- und Wertefeld angezeigt.

> **Tip**
>
> Sie können die Füllfarbe ändern, indem Sie das Farbquadrat neben der Anmerkung auswählen.

> **Note**
>
> Sie können die Schritte wiederholen um einem Diagramm mehrere vertikale Anmerkungen hinzuzufügen.

e. Um eine Anmerkung auszublenden, deaktivieren Sie das Kontrollkästchen neben der Anmerkung, die Sie im Diagramm ausblenden möchten.

f. Um eine Anmerkung zu löschen, wählen Sie das X in der Spalte Actions (Aktionen) aus.

10. Um die Y-Achse anzupassen, wählen Sie Options (Optionen). Unter Left Y-axis (Linke Y-Achse) können Sie eine benutzerdefinierte Bezeichnung für Label (Bezeichnung) eingeben. Wenn das Diagramm Werte auf der rechten Y-Achse anzeigt, können Sie auch diese Bezeichnung anpassen. Sie können auch Minimal- und Maximalwerte für die Y-Achse festlegen, so dass das Diagramm nur den von Ihnen angegebenen Wertebereich anzeigt.
11. Wenn Sie die Änderungen vorgenommen haben, wählen Sie Update Widget (Widget aktualisieren) aus.

So blenden Sie eine Diagrammlegende aus oder ändern Sie ihre Position

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie in der oberen rechten Ecke des Diagramms, das Sie bearbeiten möchten, Widget actions (Widget-Aktionen). Wählen Sie Legend (Legende) und wählen Sie Hidden (Ausgeblendet), Bottom (Unten) oder Right (Rechts).

So verbergen Sie Metriken in einem Diagramm auf einem Dashboard

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie das Farbquadrat für die Metrik aus, die Sie in der Fußzeile des Diagramms ausblenden möchten. Wenn Sie mit dem Mauszeiger über das Farbquadrat fahren, erscheint ein X, und das Quadrat wird grau, wenn Sie es auswählen.
4. Um die ausgeblendete Metrik wiederherzustellen, entfernen Sie das X im grauen Quadrat.

Original interface

So bearbeiten Sie ein Diagramm auf einem Dashboard

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Bewegen Sie den Mauszeiger über die obere rechte Ecke des Diagramms, das Sie bearbeiten möchten. Wählen Sie Widget actions (Widget-Aktionen) und anschließend Edit (Bearbeiten) aus.
4. Um den Titel des Diagramms zu ändern, wählen Sie das Bleistiftsymbol neben dem aktuellen Titel und geben Sie dann den neuen Titel ein.
5. Um den Zeitbereich des Diagramms zu ändern, wählen Sie einen der vordefinierten Zeitbereiche im oberen Bereich des Diagramms. Diese erstrecken sich von 1 Stunde bis 1 Woche (1 Std., 3 Std., 12 Std., 1 T., 3 T., oder 1 W.).
   - Um Ihren eigenen Zeitbereich festzulegen, wählen Sie custom (Benutzerdefiniert).
6. Um den Widget-Typ Ihres Diagramms zu ändern, wählen Sie die Registerkarte Graph options (Diagrammoptionen) aus. Sie können Line (Linie), Stacked area (Gestapelter Bereich), Number (Zahl), Bar (Säulen), oder Pie (Kreis) auswählen.

   > **Tip**
   >
   > Sie können den Widget-Typ Ihres Diagramms ändern, indem Sie die Dropdown-Liste neben den vordefinierten Zeitbereichen auswählen.

7. In Graphed metrics (Grafisch dargestellte Metriken) können Sie Ihrer Metrik eine dynamische Bezeichnung hinzufügen und die Bezeichnung, die Farbe der Bezeichnung, die Statistik und

den Zeitraum der Metrik ändern. Sie können auch die Position der Beschriftungen auf der Y-Achse von links nach rechts festlegen.

a. Um eine dynamische Bezeichnung für eine Metrik hinzuzufügen, wählen Sie Dynamic labels (Dynamische Labels). Dynamische Labels zeigen eine Statistik über die Metrik in der Diagrammlegende an. Dynamische Labels werden automatisch aktualisiert, wenn Ihr Dashboard oder Diagramm aktualisiert wird. Standardmäßig erscheinen die dynamischen Werte, die Sie den Etiketten hinzufügen, am Anfang der Etiketten. Weitere Informationen finden Sie unter Dynamische Labels verwenden.

b. Um die Farbe einer Metrik zu ändern, wählen Sie das Farbquadrat neben der Metrik.

c. Um die Statistik zu ändern, wählen Sie den statistischen Wert in der Spalte Statistic (Statistik) und dann einen neuen Wert aus. Weitere Informationen finden Sie unter Statistiken.

d. Um die Periode zu ändern, wählen Sie den Periodenwert in der Spalte Period (Periode) und dann einen neuen Wert aus.

8. Wenn Sie horizontale Anmerkungen hinzuzufügen oder bearbeiten möchten, wählen Sie Graph options aus:

a. Wenn Sie eine horizontale Anmerkung hinzufügen möchten, wählen Sie Add horizontal annotation aus.

b. Wählen Sie für Label (Markierung) das Stiftsymbol neben der aktuellen Anmerkung aus. Geben Sie dann Ihre neue Anmerkung ein. Wählen Sie das Häkchen aus, nachdem Sie Ihre Anmerkung eingegeben haben.

c. Wählen Sie für Value (Wert) das Stiftsymbol neben dem aktuellen Metrikwert aus. Geben Sie dann Ihren neuen Metrikwert ein. Wählen Sie nach der Eingabe Ihres Wertes das Häkchen aus.

d. Wählen Sie für Fill (Füllen) das Dropdown-Menü unter der Spalte aus und legen Sie dann fest, wie Ihre Anmerkung die Schattierung verwenden soll. Sie können None (Keine), Above (Über), Between (Zwischen) oder Below (Unter) auswählen. Wenn Sie Between (Zwischen) auswählen, wird ein neues Markierungs- und Wertefeld angezeigt.

> **Tip**
>
> Sie können die Füllfarbe ändern, indem Sie das Farbquadrat neben der Anmerkung auswählen.

e. Legen Sie für Axis (Achse) fest, ob Ihre Anmerkung auf der linken oder rechten Seite der Y-Achse angezeigt wird.

f. Um eine Anmerkung auszublenden, deaktivieren Sie das Kontrollkästchen neben der Anmerkung, die Sie im Diagramm ausblenden möchten.

g. Um eine Anmerkung zu löschen, wählen Sie das X in der Spalte Actions (Aktionen) aus.

> **Note**
>
> Sie können die Schritte wiederholen, um einem Diagramm mehrere horizontale Anmerkungen hinzuzufügen.

9. Um vertikale Anmerkungen hinzuzufügen oder zu bearbeiten, wählen Sie Graph options (Diagrammoptionen) aus:

a. Wenn Sie eine vertikale Anmerkung hinzufügen möchten, klicken Sie auf Add vertical annotation (Vertikale Anmerkung hinzufügen).

b. Wählen Sie für Label (Markierung) das Stiftsymbol neben der aktuellen Anmerkung aus. Geben Sie dann Ihre neue Anmerkung ein. Wählen Sie das Häkchen aus, nachdem Sie Ihre Anmerkung eingegeben haben.

> **Tip**
>
> Um nur Datum und Uhrzeit anzuzeigen, lassen Sie das Markierungsfeld leer.

c. Wählen Sie für Date (Datum) das Stiftsymbol neben dem aktuellen Datum und der Uhrzeit aus. Geben Sie dann das neue Datum und die neue Uhrzeit ein.

d. Wählen Sie für Fill (Füllen) das Dropdown-Menü unter der Spalte aus und legen Sie dann fest, wie Ihre Anmerkung die Schattierung verwenden soll. Sie können None (Keine), Above (Über), Between (Zwischen) oder Below (Unter) auswählen. Wenn Sie Between (Zwischen) auswählen, wird ein neues Markierungs- und Wertefeld angezeigt.

> **Tip**
>
> Sie können die Füllfarbe ändern, indem Sie das Farbquadrat neben der Anmerkung auswählen.

> **Note**
>
> Sie können die Schritte wiederholen um einem Diagramm mehrere vertikale Anmerkungen hinzuzufügen.

e. Um eine Anmerkung auszublenden, deaktivieren Sie das Kontrollkästchen neben der Anmerkung, die Sie im Diagramm ausblenden möchten.

f. Um eine Anmerkung zu löschen, wählen Sie das X in der Spalte Actions (Aktionen) aus.

10. Um die Y-Achse anzupassen, wählen Sie Graph options (Diagrammoptionen). Unter Left Y-axis (Linke Y-Achse) können Sie eine benutzerdefinierte Bezeichnung für Label (Bezeichnung) eingeben. Wenn das Diagramm Werte auf der rechten Y-Achse anzeigt, können Sie auch diese Bezeichnung anpassen. Sie können auch Minimal- und Maximalwerte für die Y-Achse festlegen, so dass das Diagramm nur den von Ihnen angegebenen Wertebereich anzeigt.
11. Wenn Sie die Änderungen vorgenommen haben, wählen Sie Update Widget (Widget aktualisieren) aus.

So blenden Sie eine Diagrammlegende aus oder ändern Sie ihre Position

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Bewegen Sie den Mauszeiger auf die obere rechte Ecke des Diagramms, das Sie bearbeiten möchten, und wählen Sie Widget actions (Widget-Aktionen). Wählen Sie Legend (Legende) und klicken Sie dann auf Hidden (Ausgeblendet), Bottom (Unten) oder Right (Rechts).

So verbergen Sie Metriken in einem Diagramm auf einem Dashboard

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie das Farbquadrat für die Metrik aus, die Sie in der Fußzeile des Diagramms ausblenden möchten. Wenn Sie mit dem Mauszeiger über das Farbquadrat fahren, erscheint ein X, und das Quadrat wird grau, wenn Sie es auswählen.

4. Um die ausgeblendete Metrik wiederherzustellen, entfernen Sie das X im grauen Quadrat.

Verschieben Sie ein Diagramm auf einem CloudWatch Dashboard oder ändern Sie die Größe

Sie können Diagramme auf Ihrem CloudWatch Dashboard anordnen und ihre Größe ändern.

So verschieben Sie ein Diagramm auf einem Dashboard

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Führen Sie eine der folgenden Aktionen aus:
   - Schieben Sie die Maus über den Diagrammtitel, bis das Auswahlsymbol erscheint. Wählen Sie das Diagramm aus und ziehen Sie es an eine neue Stelle auf dem Dashboard.
   - Um das Widget entweder in die obere linke oder die untere linke Ecke des Dashboards zu verschieben, wählen Sie die vertikale Ellipse oben rechts im Widget, um das Widget-Aktionsmenü zu öffnen. Wählen Sie dann Verschieben und wählen Sie aus, wohin das Widget verschoben werden soll.
4. Wählen Sie Save dashboard aus.

So ändern Sie die Größe eines Diagramms

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Um das Diagramm zu vergrößern oder zu verkleinern, fahren Sie mit der Maus über das Diagramm und ziehen Sie die rechte untere Ecke des Diagramms. Ziehen Sie die Ecke nicht mehr weiter, wenn Sie die gewünschte Größe haben.
4. Wählen Sie Save dashboard aus.

So vergrößern Sie ein Diagramm vorübergehend

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.

3. Wählen Sie das Diagramm aus. Fahren Sie alternativ mit der Maus über den Diagrammtitel und wählen Sie Widget actions und Enlarge aus.

Benennen Sie ein Diagramm auf einem CloudWatch Dashboard um

Sie können den Standardnamen ändern, der einem Diagramm in Ihrem Dashboard CloudWatch zugewiesen wird.

So ändern Sie auf einem Dashboard den Namen eines Diagramms

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Fahren Sie mit der Maus über den Titel des Diagramms und wählen Sie Widget actions (Widget-Aktionen) und Edit (Bearbeiten).
4. Wählen Sie oben im Bildschirm Edit graph den Titel der Grafik.
5. Geben Sie für Title (Titel) einen neuen Namen ein und wählen Sie Ok (Häkchen). Klicken Sie unten rechts auf der Seite Edit graph (Diagramm bearbeiten) auf Update widget (Widget aktualisieren).

Fügen Sie einem CloudWatch Dashboard ein Metrik-Explorer-Widget hinzu

Metrik-Explorer-Widgets enthalten Diagramme mehrerer Ressourcen mit demselben Tag oder dieselbe Ressourceneigenschaft wie einen Instance-Typ. Diese Widgets bleiben auf dem neuesten Stand, da Ressourcen, die übereinstimmen, erstellt oder gelöscht werden. Das Hinzufügen von Metrik-Explorer-Widgets zu Ihrem Dashboard hilft Ihnen, Probleme in Ihrer Umgebung effizienter zu beheben.

Beispielsweise können Sie Ihre Flotte von EC2-Instances überwachen, indem Sie Tags zuweisen, die ihre Umgebungen darstellen, z. B. Produktion oder Test. Sie können diese Tags dann verwenden, um die Betriebsmetriken wie `CPUUtilization` zu filtern und zu aggregieren, um den Zustand und die Leistung der EC2-Instances zu verstehen, die jedem Tag zugeordnet sind.

In den folgenden Schritten wird erläutert, wie Sie einem Dashboard mithilfe der Konsole ein Metrik-Explorer-Widget hinzufügen. Sie können es auch programmgesteuert oder mithilfe von hinzufügen. AWS CloudFormation Weitere Informationen finden Sie unter Objektdefinition des Metrics Explorer-Widgets und. AWS::CloudWatch::Dashboard

So fügen Sie einem Dashboard ein Metrik-Explorer-Widget hinzu

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Dashboards aus.
3. Wählen Sie den Namen des Dashboards aus, dem Sie das Metrik-Explorer-Widget hinzufügen möchten.
4. Wählen Sie das Symbol + aus.
5. Wählen Sie Explorer und danach Weiter aus.

   > **Note**
   >
   > Sie müssen sich für die neue Dashboard-Ansicht entschieden haben, um ein Metrik-Explorer-Widget hinzufügen zu können. Um sich anzumelden, wählen Sie im Navigationsbereich Dashboards und dann im Banner oben auf der Seite die neue Benutzeroberfläche ausprobieren.

6. Führen Sie eine der folgenden Aktionen aus:
   - Um eine Vorlage zu verwenden, wählen Sie Vorausgefülltes Explorer-Widget und wählen Sie dann eine zu verwendende Vorlage aus.
   - Um eine benutzerdefinierte Visualisierung zu erstellen, klicken Sie auf Leeres Explorer-Widget.
7. Wählen Sie Erstellen.

   Wenn Sie eine Vorlage verwendet haben, wird das Widget auf Ihrem Dashboard mit den ausgewählten Metriken angezeigt. Wenn Sie mit dem Explorer-Widget und dem Dashboard zufrieden sind, klicken Sie auf Speicher-Dashboard.

   Wenn Sie keine Vorlage verwendet haben, fahren Sie mit den folgenden Schritten fort.
8. Im neuen Widget unter Explorer, in der Metriken-Box eine einzelne Metrik oder alle verfügbaren Metriken aus einem Service aus.

   Nachdem Sie eine Metrik ausgewählt haben, können Sie diesen Schritt optional wiederholen, um weitere Metriken hinzuzufügen.
9. CloudWatch Zeigt für jede ausgewählte Metrik unmittelbar nach dem Metriknamen die Statistik an, die verwendet wird. Wählen Sie den Statistiknamen aus, um dies zu ändern. Wählen Sie dann die gewünschte Statistik aus.

10. Wählen Sie unter Von ein Tag oder eine Ressourceneigenschaft aus, um Ihre Ergebnisse zu filtern.

    Danach können Sie diesen Schritt optional wiederholen, um weitere Tags oder Ressourceneigenschaften auszuwählen.

    Wenn Sie mehrere Werte derselben Eigenschaft auswählen, z. B. zwei EC2-Instance-Typen, zeigt der Explorer alle Ressourcen an, die mit den ausgewählten Eigenschaften übereinstimmen. Es wird als ODER-Operation behandelt.

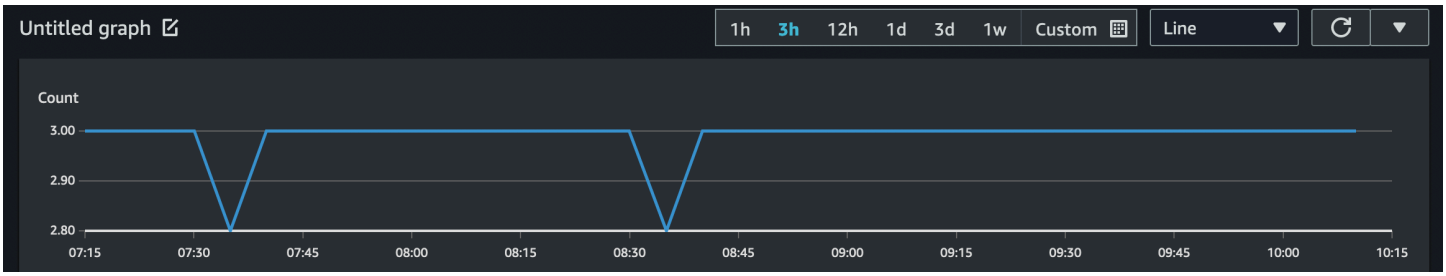
    Wenn Sie andere Eigenschaften oder Tags auswählen, z. B. das **`Production`**-Tag und den M5-Instance-Typ, werden nur die Ressourcen angezeigt, die allen diesen Auswahlen entsprechen. Dies wird als UND-Operation behandelt.
11. (Optional) Wählen Sie für Aggregieren nach eine Statistik aus, die zum Aggregieren der Metriken verwendet werden soll. Wählen Sie dann neben für aus, wie die Metrik aus der Liste aggregiert werden soll. Sie können alle Ressourcen zusammenfassen, die derzeit angezeigt werden, oder durch ein einzelnes Tag oder eine Ressourceneigenschaft aggregieren.

    Je nachdem, wie Sie sich für die Aggregierung entscheiden, kann das Ergebnis eine einzelne Zeitreihe oder mehrere Zeitreihen sein.
12. Unter Aufteilen nach können Sie ein einzelnes Diagramm mit mehreren Zeitreihen in mehrere Diagramme aufteilen. Die Aufteilung kann nach verschiedenen Kriterien erfolgen, die Sie unter Aufteilen nach auswählen.
13. Unter Diagrammoptionen können Sie das Diagramm verfeinern, indem Sie den Zeitraum, den Diagrammtyp, die Legendenplatzierung und das Layout ändern.
14. Wenn Sie mit dem Explorer-Widget und dem Dashboard zufrieden sind, klicken Sie auf Dashboard speichern.

Fügen Sie ein Linien-Widget zu einem Dashboard hinzu oder entfernen Sie CloudWatch es

Mit dem Linien-Widget können Sie Kennzahlen über einen bestimmten Zeitraum hinweg vergleichen. Sie können auch das Mini-Map-Zoom-Feature des Widgets verwenden, um Abschnitte von Liniendiagrammen zu untersuchen, ohne zwischen vergrößerter und verkleinerter Ansicht zu wechseln. Die Verfahren in diesem Abschnitt beschreiben, wie Sie ein Linien-Widget in einem CloudWatch Dashboard hinzufügen und entfernen. Informationen zur Verwendung des Mini-Map-

Zoom-Features des Widgets mit Liniendiagrammen finden Sie unter Vergrößern eines Linien- oder gestapelten Flächendiagramms.



So fügen Sie ein Linien-Widget zu einem Dashboard hinzu

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie das Symbol + aus, und klicken Sie auf Line (Linie).
4. Wählen Sie Metrics (Metriken) aus.
5. Klicken Sie auf Browse (Durchsuchen) und wählen Sie die Metrik aus, die Sie grafisch darstellen möchten.
6. Klicken Sie auf Create widget (Widget erstellen) und wählen Sie dann Save dashboard (Dashboard speichern) aus.

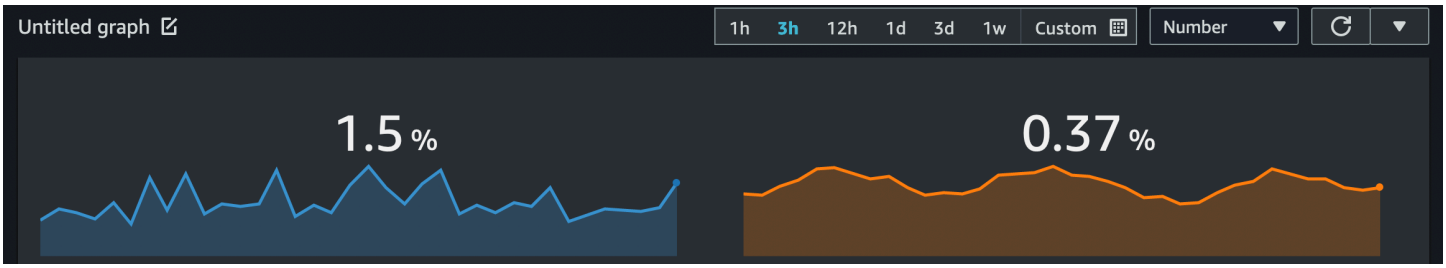
So entfernen Sie ein Linien-Widget aus einem Dashboard

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie in der oberen rechten Ecke des Zeilen-Widgets, das Sie entfernen möchten, Widget actions (Widget-Aktionen) und dann Delete (Löschen).
4. Wählen Sie Save dashboard aus.

Fügen Sie ein Zahlen-Widget zu einem CloudWatch Dashboard hinzu oder entfernen Sie es

Mit dem Zahlen-Widget können Sie sich die neuesten Metrikwerte und Trends ansehen, sobald sie erscheinen. Da das Zahlen-Widget das Sparkline-Feature enthält, können Sie die obere und untere Hälfte der metrischen Trends in einem einzigen Diagramm visualisieren. Die Verfahren in diesem

Abschnitt beschreiben, wie Sie ein Zahlen-Widget zu einem CloudWatch Dashboard hinzufügen und daraus entfernen.



> **Note**
>
> Nur das neue Interface unterstützt das Sparkline-Feature. Wenn Sie ein Zahlen-Widget erstellen, wird das Sparkline-Feature automatisch eingeschlossen.

So fügen Sie einem Dashboard ein Zahlen-Widget hinzu

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie das Symbol + aus, und klicken Sie auf Number (Nummer).
4. Suchen Sie auf der Registerkarte Durchsuchen nach der Metrik, die Sie anzeigen möchten.
5. (Optional) Um die Farbe des Sparkline-Features zu ändern, wählen Sie Grafisch dargestellte Metriken und markieren Sie das Farbkästchen neben der Kennzeichnung der Metrik. Es erscheint ein Menü, in dem Sie eine andere Farbe auswählen oder einen sechsstelligen Hex-Farbcode eingeben können, um eine Farbe festzulegen.
6. (Optional) Um das Sparkline-Feature auszuschalten, wählen Sie Optionen. Unter Sparkline, das Kontrollkästchen.
7. (Optional) Um den Zeitbereich Ihres Zahlen-Widgets zu ändern, wählen Sie einen der vordefinierten Zeitbereiche im oberen Bereich des Widgets. Die Zeitspanne reicht von 1 Stunde bis 1 Woche (1h, 3h, 12h, 1d, 3d, oder 1w).

   Um Ihren eigenen Zeitraum festzulegen, wählen Sie Custom (Benutzerdefiniert).

   - (Optional) Damit dieses Widget weiterhin den von Ihnen ausgewählten Zeitraum verwendet, auch wenn der Zeitraum für den Rest des Dashboards später geändert wird, wählen Sie Zeitraum beibehalten.
8. (Optional) Damit das Zahlen-Widget ein Aggregat anzeigt (1h, 3h , 12h, 1d, 3d oder 1w).

Um Ihren eigenen Zeitraum festzulegen, wählen Sie Custom (Benutzerdefiniert).

- (Optional) Damit dieses Widget anstelle des neuesten Werts einen Durchschnitt des Metrikwerts über den gesamten Zeitraum anzeigt, wählen Sie Optionen. Der Zeitbereichswert zeigt den Wert aus dem gesamten Zeitraum an.

9. Klicken Sie auf Create widget (Widget erstellen) und wählen Sie Save dashboard (Dashboard speichern).

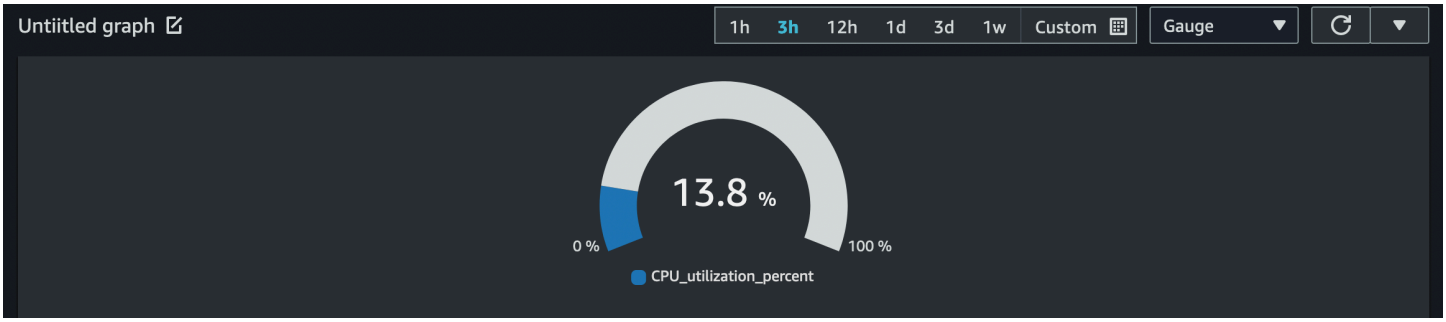
> **Tip**
>
> Sie können das Sparkline-Feature über das Zahlen-Widget auf dem Dashboard-Bildschirm deaktivieren. Wählen Sie in der rechten oberen Ecke des Zahlen-Widgets, das Sie ändern möchten, Widget actions (Widget-Aktionen). Wählen Sie Sparkline und dann wählen Sie Hide sparkline (Sparkline ausblenden).

So entfernen Sie ein Zahlen-Widget von einem Dashboard

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards, und wählen Sie dann das Dashboard aus, das das zu löschende Zahlen-Widget enthält.
3. Wählen Sie in der oberen rechten Ecke des Zahlen-Widgets, das Sie entfernen möchten, Widget actions (Widget-Aktionen) und dann Delete (Löschen).
4. Wählen Sie Save dashboard aus.

Fügen Sie ein Messgerät-Widget zu einem CloudWatch Dashboard hinzu oder entfernen Sie es

Mit dem Messinstrument-Widget können Sie Metrikwerte visualisieren, die zwischen Bereichen liegen. Sie können beispielsweise das Messinstrument-Widget verwenden, um Prozentsätze und CPU-Auslastung darzustellen, damit Sie auftretende Leistungsprobleme beobachten und diagnostizieren können. Die Verfahren in diesem Abschnitt beschreiben, wie Sie ein Messgerät-Widget zu einem CloudWatch Dashboard hinzufügen und daraus entfernen.



> **Note**
>
> Nur die neue Oberfläche in der CloudWatch Konsole unterstützt die Erstellung des Messgerät-Widgets. Sie müssen einen Messbereich festlegen, wenn Sie dieses Widget erstellen.

So fügen Sie ein Messinstrument-Widget zu einem Dashboard hinzu

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie auf dem Dashboard-Bildschirm das Symbol + und dann Gauge (Messinstrument).
4. Klicken Sie auf Browse (Durchsuchen) und wählen Sie dann die Metrik aus, die Sie grafisch darstellen möchten.
5. Wählen Sie Optionen aus. Stellen Sie unter Gauge range (Messinstrumentbereich) die Werte für Min und Max ein. Für Prozentsätze wie CPU-Auslastung empfehlen wir, die Werte für `Min` auf `0` und `Max` auf `100` festzulegen.
6. (Optional) Um die Farbe des Messinstrument-Widgets zu ändern, wählen Sie Graphed metrics (Grafisch dargestellte Metriken) und aktivieren Sie das Farbkästchen neben der Metrikbeschriftung. Es erscheint ein Menü, in dem Sie eine andere Farbe auswählen oder einen sechsstelligen Hex-Farbcode eingeben können, um eine Farbe festzulegen.
7. Klicken Sie auf Create widget (Widget erstellen) und wählen Sie Save dashboard (Dashboard speichern).

So entfernen Sie ein Messinstrument-Widget aus einem Dashboard

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

2. Wählen Sie im Navigationsbereich die Option Dashboards, und wählen Sie dann das Dashboard aus, das das zu löschende Messinstrument-Widget enthält.
3. Wählen Sie in der oberen rechten Ecke des Messinstrument-Widgets, das Sie löschen möchten, Widget actions (Widget-Aktionen) und wählen Sie Delete (Löschen).
4. Wählen Sie Save dashboard aus.

Fügen Sie einem CloudWatch Dashboard ein benutzerdefiniertes Widget hinzu

Ein benutzerdefiniertes Widget ist ein CloudWatch Dashboard-Widget, das jede AWS Lambda Funktion mit benutzerdefinierten Parametern aufrufen kann. Anschließend wird das zurückgegebene HTML oder JSON angezeigt. Benutzerdefinierte Widgets sind eine einfache Möglichkeit, eine benutzerdefinierte Datenansicht auf einem Dashboard zu erstellen. Wenn Sie Lambda-Code schreiben und HTML erstellen können, können Sie ein nützliches benutzerdefiniertes Widget erstellen. Darüber hinaus bietet Amazon mehrere vorgefertigte benutzerdefinierte Widgets, die Sie ohne Code erstellen können.

Wenn Sie eine Lambda-Funktion erstellen, die als benutzerdefiniertes Widget verwendet werden soll, empfehlen wir dringend, das Präfix customWidget in den Funktionsnamen aufzunehmen. Auf diese Weise können Sie wissen, welche Ihrer Lambda-Funktionen sicher zu verwenden sind, wenn Sie benutzerdefinierte Widgets zu Ihrem Dashboard hinzufügen.

Benutzerdefinierte Widgets verhalten sich wie andere Widgets auf Ihrem Dashboard. Sie können aktualisiert und automatisch aktualisiert, in der Größe geändert und verschoben werden. Sie reagieren auf den Zeitbereich des Dashboards.

Wenn Sie kontoübergreifende CloudWatch Konsolenfunktionen eingerichtet haben, können Sie ein in einem Konto erstelltes benutzerdefiniertes Widget zu Dashboards in anderen Konten hinzufügen. Weitere Informationen finden Sie unter Kontoübergreifende, regionsübergreifende Konsole CloudWatch .

Sie können benutzerdefinierte Widgets auch auf Ihrer eigenen Website verwenden, indem Sie die Funktion zum Teilen von CloudWatch Dashboards verwenden. Weitere Informationen finden Sie unter CloudWatch Dashboards teilen.

Themen

- Details zu benutzerdefinierten Widgets

- Sicherheit und JavaScript
- Interaktivität im benutzerdefinierten Widget
- Erstellen eines benutzerdefinierten Widgets
- Beispiel-Widgets

Details zu benutzerdefinierten Widgets

Folgende Funktionen sind zulässig:

1. Das CloudWatch Dashboard ruft die Lambda-Funktion auf, die den Widget-Code enthält. Es übergibt alle benutzerdefinierten Parameter, die im Widget definiert sind.
2. Die Lambda-Funktion gibt eine Zeichenfolge von HTML, JSON oder Markdown zurück. Markdown wird als JSON im folgenden Format zurückgegeben:

   ```
   {"markdown":"markdown content"}
   ```

3. Das Dashboard zeigt das zurückgegebene HTML oder JSON an.

Wenn die Funktion HTML zurückgibt, werden die meisten HTML-Tags unterstützt. Sie können Cascading Style Sheets (CSS)-Stile und Scalable Vector Graphics (SVG) verwenden, um anspruchsvolle Ansichten zu erstellen.

Der Standardstil von HTML-Elementen wie Links und Tabellen folgt dem Stil von CloudWatch Dashboards. Sie können diesen Stil mithilfe von Inline-Stilen anpassen, indem Sie das `<style>`-Tag verwenden. Sie können die Standardstile auch deaktivieren, indem Sie ein einzelnes HTML-Element in die Klasse `cwdb-no-default-styles` aufnehmen. Im folgenden Beispiel werden Standardstile deaktiviert: `<div class="cwdb-no-default-styles"></div>`.

Jeder Aufruf eines benutzerdefinierten Widgets an Lambda enthält ein `widgetContext`-Element mit den folgenden Inhalten, um dem Entwickler der Lambda-Funktion nützliche Kontextinformationen bereitzustellen.

```
{
    "widgetContext": {
        "dashboardName": "Name-of-current-dashboard",
        "widgetId": "widget-16",
        "accountId": "012345678901",
```

```
        "locale": "en",
        "timezone": {
            "label": "UTC",
            "offsetISO": "+00:00",
            "offsetInMinutes": 0
        },
        "period": 300,
        "isAutoPeriod": true,
        "timeRange": {
            "mode": "relative",
            "start": 1627236199729,
            "end": 1627322599729,
            "relativeStart": 86400012,
            "zoom": {
                "start": 1627276030434,
                "end": 1627282956521
             }
        },
        "theme": "light",
        "linkCharts": true,
        "title": "Tweets for Amazon website problem",
        "forms": {
            "all": {}
        },
        "params": {
            "original": "param-to-widget"
        },
        "width": 588,
        "height": 369
    }
}
```

Standard-CSS-Styling

Benutzerdefinierte Widgets bieten die folgenden Standard-CSS-Styling-Elemente:

- Sie können die CSS-Klasse btn verwenden, um eine Schaltfläche hinzuzufügen. Es verwandelt einen Anker (<a>) in eine Schaltfläche wie im folgenden Beispiel:

  ```
  <a class="btn" href=https://amazon.com”>Open Amazon</a>
  ```

- Sie können die CSS-Klasse btn btn-primary verwenden, um eine primäre Schaltfläche hinzuzufügen.

- Die folgenden Elemente werden standardmäßig formatiert: Tabelle, Auswahl, Header (h1, h2 und h3), vorformatierter Text (pre), Eingabe und Textbereich.

Verwenden des Beschreibungs-Parameters

Wir empfehlen Ihnen dringend, den Parameter Beschreiben in Ihren Funktionen zu unterstützen, auch wenn er nur einen leeren String zurückgibt. Wenn Sie es nicht unterstützen und es im benutzerdefinierten Widget aufgerufen wird, zeigt es Widget-Inhalt an, als wäre es Dokumentation.

Wenn Sie den Parameter Beschreiben einschließen, gibt die Lambda-Funktion die Dokumentation im Markdown-Format zurück und tut nichts anderes.

Wenn Sie in der Konsole ein benutzerdefiniertes Widget erstellen, wird nach Auswahl der Lambda-Funktion die Schaltfläche Dokumentation abrufen angezeigt. Wenn Sie diese Schaltfläche wählen, wird die Funktion mit dem Parameter Beschreiben aufgerufen und die Dokumentation der Funktion zurückgegeben. Wenn die Dokumentation in Markdown gut formatiert ist, CloudWatch analysiert sie den ersten Eintrag in der Dokumentation, der in YAML von drei einzelnen Backtick-Zeichen (```) umgeben ist. Anschließend wird die Dokumentation automatisch in den Parametern aufgefüllt. Nachstehend finden Sie ein Beispiel für eine gut formatierte Dokumentation.

```
``` yaml
echo: <h1>Hello world</h1>
```
```

Sicherheit und JavaScript

Aus Sicherheitsgründen JavaScript ist dies im zurückgegebenen HTML-Code nicht zulässig. Durch das Entfernen von werden Probleme mit der Eskalation von Berechtigungen JavaScript verhindert, bei denen der Autor der Lambda-Funktion Code einfügt, der mit höheren Berechtigungen ausgeführt werden könnte als der Benutzer, der das Widget im Dashboard betrachtet.

Wenn der zurückgegebene JavaScript HTML-Code Code oder andere bekannte Sicherheitslücken enthält, wird er aus dem HTML-Code entfernt, bevor er auf dem Dashboard gerendert wird. Beispielsweise sind die Tags <iframe> und <use> nicht zulässig und werden entfernt.

Benutzerdefinierte Widgets werden standardmäßig nicht in einem Dashboard ausgeführt. Stattdessen müssen Sie explizit die Ausführung eines benutzerdefinierten Widgets zulassen, wenn Sie der aufrufenden Lambda-Funktion vertrauen. Sie können wählen, ob Sie es einmal zulassen oder immer

zulassen, sowohl für einzelne Widgets als auch für das gesamte Dashboard. Sie können auch die Berechtigung für einzelne Widgets und das gesamte Dashboard verweigern.

Interaktivität im benutzerdefinierten Widget

Auch wenn dies nicht zulässig JavaScript ist, gibt es andere Möglichkeiten, Interaktivität mit dem zurückgegebenen HTML-Code zu ermöglichen.

- Jedes Element im zurückgegebenen HTML kann mit einer speziellen Konfiguration in einem `<cwdb-action>`-Tag versehen werden, das Informationen in Pop-ups anzeigen, bei Klicks um Bestätigung bitten und jede Lambda-Funktion aufrufen kann, wenn dieses Element ausgewählt wird. Sie können beispielsweise Schaltflächen definieren, die eine beliebige AWS API mithilfe einer Lambda-Funktion aufrufen. Der zurückgegebene HTML-Code kann so eingestellt werden, dass er entweder den Inhalt des vorhandenen Lambda-Widgets ersetzt oder in einem Modal angezeigt wird.
- Der zurückgegebene HTML-Code kann Links enthalten, die neue Konsolen öffnen, andere Kundenseiten öffnen oder andere Dashboards laden.
- Der HTML-Code kann das `title`-Attribut für ein Element enthalten, das zusätzliche Informationen liefert, wenn der Benutzer den Mauszeiger über dieses Element bewegt.
- Das Element kann CSS-Selektoren wie `:hover` enthalten, die Animationen oder andere CSS-Effekte aufrufen können. Sie können auch Elemente auf der Seite ein- oder ausblenden.

<cwdb-action> Definition und Verwendung

Das `<cwdb-action>`-Element definiert ein Verhalten für das unmittelbar vorhergehende Element. Der Inhalt von `<cwdb-action>` ist entweder HTML zum Anzeigen oder ein JSON-Parameterblock, der an eine Lambda-Funktion übergeben wird.

Es folgt ein Beispiel eines `<cwdb-action>`-Elements:

```
<cwdb-action
     action="call|html"
     confirmation="message"
     display="popup|widget"
     endpoint="<lambda ARN>"
     event="click|dblclick|mouseenter">

     html | params in JSON
</cwdb-action>
```

- Aktion – Gültige Werte sind `call`, das eine Lambda-Funktion aufruft, und `html`, das jegliches HTML anzeigt, das in `<cwdb-action>` enthalten ist. Der Standardwert ist `html`.
- Bestätigung – Zeigt eine Bestätigungsnachricht an, die bestätigt werden muss, bevor die Aktion ausgeführt wird, damit der Kunde stornieren kann.
- Anzeige – Gültige Werte sind popup und `widget`, die den Inhalt des Widgets selbst ersetzen. Der Standardwert ist `widget`.
- Endpunkt – Der Amazon-Ressourcenname (ARN) der aufzurufenden Lambda-Funktion. Dies ist erforderlich, wenn `action call` ist.
- -Ereignis – Definiert das Ereignis auf dem vorherigen Element, das die Aktion aufruft. Gültige Werte sind `click`, `dblclick` und `mouseenter`. Das Ereignis `mouseenter` kann nur in Kombination mit der Aktion `html` verwendet werden. Der Standardwert ist `click`.

Beispiele

Im Folgenden finden Sie ein Beispiel für die Verwendung des `<cwdb-action>`-Tags zum Erstellen einer Schaltfläche, die eine Amazon-EC2-Instance mithilfe eines Lambda-Funktionsaufrufs neu startet. Es zeigt den Erfolg oder das Fehlschlagen des Anrufs in einem Pop-up an.

```
<a class="btn">Reboot Instance</a>
<cwdb-action action="call" endpoint="arn:aws:lambda:us-
east-1:123456:function:rebootInstance" display="popup">
       { "instanceId": "i-342389adbfef" }
</cwdb-action>
```

Im nächsten Beispiel werden weitere Informationen in einem Pop-up angezeigt.

```
<a>Click me for more info in popup</a>
<cwdb-action display="popup">
   <h1>Big title</h1>
   More info about <b>something important</b>.
</cwdb-action>
```

Dieses Beispiel ist eine Schaltfläche Weiter, die den Inhalt eines Widgets durch einen Aufruf einer Lambda-Funktion ersetzt.

```
<a class="btn btn-primary">Next</a>
<cwdb-action action="call" endpoint="arn:aws:lambda:us-
east-1:123456:function:nextPage">
```

```
   { "pageNum": 2 }
</cwdb-action>
```

Erstellen eines benutzerdefinierten Widgets

Um ein benutzerdefiniertes Widget zu erstellen, können Sie eines der von AWS bereitgestellten Beispiele verwenden oder ein eigenes erstellen. Die AWS Beispiele enthalten sowohl Beispiele in Python als auch in Python JavaScript und werden von einem AWS CloudFormation Stack erstellt. Eine Liste von Beispielen finden Sie unter Beispiel-Widgets.

Um ein benutzerdefiniertes Widget in einem CloudWatch Dashboard zu erstellen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie das Symbol + aus.
4. Klicken Sie auf Benutzerdefiniertes Widget.
5. Verwenden Sie eine der folgenden Methoden:
   - Gehen Sie wie folgt vor AWS, um ein von bereitgestelltes benutzerdefiniertes Beispiel-Widget zu verwenden:
     a. Wählen Sie das Muster in der Dropdown-Box aus.

        Die AWS CloudFormation Konsole wird in einem neuen Browser gestartet. Gehen Sie in der AWS CloudFormation Konsole wie folgt vor:

     b. (Optional) Passen Sie den AWS CloudFormation Stack-Namen an.
     c. Treffen Sie eine Auswahl für alle Parameter, die von der Probe verwendet werden.
     d. Wählen Sie Ich bestätige, dass AWS CloudFormation möglicherweise IAM-Ressourcen erstellt werden, und wählen Sie Stack erstellen aus.
   - Gehen Sie wie folgt vor, um Ihr eigenes benutzerdefiniertes Widget zu erstellen AWS, das von bereitgestellt wird:
     a. Wählen Sie Weiter aus.
     b. Wählen Sie entweder aus einer Liste Ihre Lambda-Funktion aus oder geben Sie ihren Amazon-Ressourcennamen (ARN) ein. Wenn Sie sie aus einer Liste auswählen, geben Sie auch die Region an, in der sich die Funktion befindet, und die zu verwendende Version.

c. Treffen Sie unter Parameter eine Auswahl für alle Parameter, die von der Funktion verwendet werden.

d. Geben Sie einen Titel für das Widget ein.

e. Konfigurieren Sie für Update auf, wann das Widget aktualisiert werden soll (wann die Lambda-Funktion erneut aufgerufen werden soll). Dies kann eine oder mehrere der folgenden sein: Aktualisieren, um es zu aktualisieren, wenn das Dashboard automatisch aktualisiert wird, Größe ändern, um es bei jeder Größenänderung des Widgets zu aktualisieren, oder Zeitbereich, um es zu aktualisieren, wenn der Zeitbereich des Dashboards angepasst wird, einschließlich beim Zoomen von Diagrammen hinein.

f. Wenn Sie mit der Vorversion zufrieden sind, wählen Sie Widget erstellen.

Beispiel-Widgets

AWS bietet Beispiele für benutzerdefinierte Widgets sowohl in Python als JavaScript auch in Python. Sie können diese Beispiel-Widgets erstellen, indem Sie den Link für jedes Widget in dieser Liste verwenden. Alternativ können Sie ein Widget mithilfe der CloudWatch Konsole erstellen und anpassen. Die Links in dieser Liste öffnen eine AWS CloudFormation Konsole und verwenden einen AWS CloudFormation Schnellerstellungslink, um das benutzerdefinierte Widget zu erstellen.

Sie können auf die Beispiele für benutzerdefinierte Widgets auch unter zugreifen. GitHub

Im Anschluss an diese Liste werden für jede Sprache vollständige Beispiele des Echo-Widgets angezeigt.

JavaScript

Beispiele für benutzerdefinierte Widgets finden Sie in JavaScript

- echo – Ein einfaches Echo, mit dem Sie testen können, wie HTML in einem benutzerdefinierten Widget angezeigt wird, ohne ein neues Widget schreiben zu müssen.
- Hello World – Ein sehr einfaches Starter-Widget.
- Debugger für benutzerdefinierte Widgets – Ein Debugger-Widget, das nützliche Informationen zur Lambda-Laufzeitumgebung anzeigt.
- CloudWatch Logs Insights abfragen — CloudWatch Logs Insights-Abfragen ausführen und bearbeiten.
- Ausführen von Amazon-Athena-Abfragen – Ausführen und Bearbeiten von Athena-Abfragen.

- AWS API aufrufen — Rufen Sie eine beliebige schreibgeschützte AWS API auf und zeigen Sie die Ergebnisse im JSON-Format an.
- Schnelles CloudWatch Bitmap-Diagramm — Rendern Sie CloudWatch Diagramme serverseitig für eine schnelle Anzeige.
- Text-Widget aus dem CloudWatch Dashboard — Zeigt das erste Text-Widget aus dem angegebenen CloudWatch Dashboard an.
- CloudWatch Metrikdaten als Tabelle — Zeigt CloudWatch metrische Rohdaten in einer Tabelle an.
- Amazon-EC2-Tabelle – Zeigt die wichtigsten EC2-Instances nach CPU-Auslastung an. Dieses Widget enthält auch eine Schaltfläche „Neustart“, die standardmäßig deaktiviert ist.
- AWS CodeDeploy Bereitstellungen — Zeigt CodeDeploy Bereitstellungen an.
- AWS Cost Explorer Bericht — Zeigt einen Bericht über die Kosten der einzelnen AWS Dienste für einen ausgewählten Zeitraum an.
- Inhalt der externen URL anzeigen – Zeigt den Inhalt einer extern zugänglichen URL an.
- Anzeigen eines Amazon-S3-Objekts – Zeigt ein Objekt in einem Amazon-S3-Bucket in Ihrem Konto an.
- Einfaches SVG-Kreisdiagramm – Beispiel für ein grafisches SVG-basiertes Widget.

Python

Beispiel für benutzerdefinierte Widgets in Python

- echo – Ein einfaches Echo, mit dem Sie testen können, wie HTML in einem benutzerdefinierten Widget angezeigt wird, ohne ein neues Widget schreiben zu müssen.
- Hello World – Ein sehr einfaches Starter-Widget.
- Debugger für benutzerdefinierte Widgets – Ein Debugger-Widget, das nützliche Informationen zur Lambda-Laufzeitumgebung anzeigt.
- AWS API aufrufen — Rufen Sie eine beliebige schreibgeschützte AWS API auf und zeigen Sie die Ergebnisse im JSON-Format an.
- Schnelles CloudWatch Bitmap-Diagramm — Rendern Sie CloudWatch Diagramme serverseitig für eine schnelle Anzeige.
- Dashboard-Snapshot per E-Mail senden – Erstellen Sie einen Snapshot des aktuellen Dashboards und senden Sie es an E-Mail-Empfänger.

- Senden Sie ein Dashboard-Snapshot an Amazon S3 – Erstellen Sie einen Snapshot des aktuellen Dashboards und speichern Sie ihn in Amazon S3.
- Text-Widget aus dem CloudWatch Dashboard — Zeigt das erste Text-Widget aus dem angegebenen CloudWatch Dashboard an.
- Inhalt der externen URL anzeigen – Zeigt den Inhalt einer extern zugänglichen URL an.
- RSS-Reader – Zeigt RSS-Feeds an.
- Anzeigen eines Amazon-S3-Objekts – Zeigt ein Objekt in einem Amazon-S3-Bucket in Ihrem Konto an.
- Einfaches SVG-Kreisdiagramm – Beispiel für ein grafisches SVG-basiertes Widget.

Echo-Widget in JavaScript

Im Folgenden finden Sie das Echo-Beispiel-Widget in JavaScript.

```
const DOCS = `
## Echo
A basic echo script. Anything passed in the \`\`\`echo\`\`\` parameter is returned as
 the content of the custom widget.
### Widget parameters
Param | Description
---|---
**echo** | The content to echo back

### Example parameters
\`\`\` yaml
echo: <h1>Hello world</h1>
\`\`\`
`;

exports.handler = async (event) => {
    if (event.describe) {
        return DOCS;
    }

    let widgetContext = JSON.stringify(event.widgetContext, null, 4);
    widgetContext = widgetContext.replace(/</g, '&lt;');
    widgetContext = widgetContext.replace(/>/g, '&gt;');

    return `${event.echo || ''}<pre>${widgetContext}</pre>`;
};
```

Echo-Widget in Python

Im Folgenden finden Sie das Echo-Beispiel-Widget in Python.

```
import json

DOCS = """
## Echo
A basic echo script. Anything passed in the ```echo``` parameter is returned as the
 content of the custom widget.
### Widget parameters
Param | Description
---|---
**echo** | The content to echo back

### Example parameters
``` yaml
echo: <h1>Hello world</h1>
```"""

def lambda_handler(event, context):
    if 'describe' in event:
        return DOCS

    echo = event.get('echo', '')
    widgetContext = event.get('widgetContext')
    widgetContext = json.dumps(widgetContext, indent=4)
    widgetContext = widgetContext.replace('<', '<')
    widgetContext = widgetContext.replace('>', '>')

    return f'{echo}<pre>{widgetContext}</pre>'
```

Echo-Widget in Java

Im Folgenden finden Sie das Echo-Beispiel-Widget in Java.

```
package example;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;

import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
```

```
public class Handler implements RequestHandler<Event, String>{

  static String DOCS = ""
    + "## Echo\n"
    + "A basic echo script. Anything passed in the ```echo``` parameter is returned as
 the content of the custom widget.\n"
    + "### Widget parameters\n"
    + "Param | Description\n"
    + "---|---\n"
    + "**echo** | The content to echo back\n\n"
    + "### Example parameters\n"
    + "```yaml\n"
    + "echo: <h1>Hello world</h1>\n"
    + "```\n";

  Gson gson = new GsonBuilder().setPrettyPrinting().create();

  @Override
  public String handleRequest(Event event, Context context) {

    if (event.describe) {
      return DOCS;
    }

    return (event.echo != null ? event.echo : "") + "<pre>" +
 gson.toJson(event.widgetContext) + "</pre>";
  }
}

class Event {

    public boolean describe;
    public String echo;
    public Object widgetContext;

    public Event() {}

    public Event(String echo, boolean describe, Object widgetContext) {
        this.describe = describe;
        this.echo = echo;
        this.widgetContext = widgetContext;
    }
```

```
}
```

Fügen Sie ein Text-Widget zu einem CloudWatch Dashboard hinzu oder entfernen Sie es

Ein Text-Widget enthält einen Textblock im Format Markdown. Sie können Text-Widgets zu Ihrem CloudWatch Dashboard hinzufügen, bearbeiten oder daraus entfernen.

So fügen Sie einem Dashboard ein Text-Widget hinzu

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie das Symbol + aus.
4. Wählen Sie Text.
5. Fügen Sie für Markdown Ihren Text hinzu und formatieren Sie ihn mithilfe von Markdown und klicken Sie anschließend auf Create widget (Widget erstellen).
6. Um das Text-Widget transparent zu machen, wählen Sie Transparenter Hintergrund.
7. Wählen Sie Save dashboard aus.

So bearbeiten Sie ein Text-Widget auf einem Dashboard

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Fahren Sie mit der Maus über die obere rechte Ecke des Textblocks und klicken Sie auf Widget actions (Widget-Aktionen). Wählen Sie dann Edit (Bearbeiten) aus.
4. Aktualisieren Sie den Text bei Bedarf und klicken Sie dann auf Update widget (Widget aktualisieren).
5. Wählen Sie Save dashboard aus.

So entfernen Sie ein Text-Widget von einem Dashboard

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Fahren Sie mit der Maus über die obere rechte Ecke des Textblocks und klicken Sie auf Widget actions (Widget-Aktionen). Wählen Sie dann Löschen aus.

4. Wählen Sie Save dashboard aus.

Fügen Sie ein Alarm-Widget zu einem CloudWatch Dashboard hinzu oder entfernen Sie es

Um einem Dashboard ein Alarm-Widget hinzuzufügen, wählen Sie eine der folgenden Optionen aus:

- Sie können einen einzelnen Alarm in einem Widget hinzufügen, der sowohl das Diagramm der Metrik des Alarms als auch den Alarmstatus anzeigt.
- Sie können ein Alarmstatus-Widget hinzufügen, das den Status mehrerer Alarme in einem Raster anzeigt. Es werden nur die Alarmnamen und der aktuelle Status angezeigt, die Diagramme werden nicht angezeigt. Bis zu 100 Alarme können in einem Alarmstatus-Widget einbezogen werden.

So fügen Sie einem Dashboard einen einzelnen Alarm einschließlich des zugehörigen Diagramms hinzu

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Alarms, wählen Sie den hinzuzufügenden Alarm aus und dann Add to Dashboard.
3. Wählen Sie ein Dashboard aus, wählen Sie einen Widget-Typ (Line, Stacked area, or Number) und wählen Sie dann Add to Dashboard.
4. Um den Alarm auf dem Dashboard zu sehen, wählen Sie im Navigationsbereich Dashboards und wählen Sie dann das Dashboard aus.
5. (Optional) Um ein Alarmdiagramm vorübergehend größer darzustellen, wählen Sie das Diagramm aus.
6. (Optional) Um den Widget-Typ zu ändern, bewegen Sie den Mauszeiger über den Titel des Diagramms, wählen Sie Widget-Aktionen und dann Widget-Typ.

So fügen Sie einem Dashboard ein Alarmstatus-Widget hinzu

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie das Symbol + aus.
4. Klicken Sie auf Alarmstatus.

5. Aktivieren Sie die Kontrollkästchen neben den Alarmen, die Sie dem Widget hinzufügen möchten, und wählen Sie dann Widget erstellen aus.
6. Wählen Sie Add to dashboard (Zu Dashboard hinzufügen) aus.

So entfernen Sie ein Alarm-Widget von einem Dashboard

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Bewegen Sie den Mauszeiger über das Widget, wählen Sie Widget-Aktionen und wählen Sie dann Löschen.
4. Wählen Sie Save dashboard aus. Wenn Sie versuchen, den Navigationsbereich des Dashboards zu verlassen, bevor Sie Ihre Änderungen gesichert haben, werden Sie aufgefordert, Ihre Änderungen entweder zu speichern oder zu verwerfen.

Fügen Sie ein Datentabellen-Widget zu einem CloudWatch Dashboard hinzu oder entfernen Sie es

Mit dem Datentabellen-Widget können Sie die Rohdatenpunkte Ihrer Metrik und eine kurze Zusammenfassung dieser Rohdaten sehen. Da es sich beim Datentabellen-Widget nicht um ein Diagramm handelt, das die tatsächlichen Daten von Ihnen abstrahiert, ist es einfacher, die dargestellten Datenpunkte zu verstehen. Die Verfahren in diesem Abschnitt beschreiben, wie Sie ein Datentabellen-Widget zu einem CloudWatch Dashboard hinzufügen und daraus entfernen.

Tabelleneigenschaften

Eine Datentabelle hat einen Standardsatz von Eigenschaften, für die keine Änderungen an den Optionen oder der Quelle erforderlich sind. Zu diesen Eigenschaften gehören, dass die Label-Spalte fixiert ist, alle Übersichtsspalten aktiviert sind, Datenpunkte gerundet sind und ihre Einheiten umgerechnet wurden.

Jedes Datentabellen-Widget kann die folgenden Eigenschaften haben. Zu den Informationen zu den einzelnen Eigenschaften gehört auch, wie sie in der JSON-Quelle des Dashboards konfiguriert werden. Weitere Informationen über Dashboard-JSON finden Sie unter Textaufbau und Syntax des Dashboards.

Übersicht

Übersichtsspalten sind eine neue Eigenschaft, die mit dem Datentabellen-Widget eingeführt wurde. Bei diesen Spalten handelt es sich um eine bestimmte Teilmenge von Zusammenfassungen Ihrer aktuellen Tabelle. Die Sum-Zusammenfassung ist beispielsweise eine Summe aller angezeigten Datenpunkte in ihrer Zeile. Die Übersichtsspalten sind nicht mit CloudWatch Statistiken identisch. In der Quelle dargestellt als:

```
"table": {
       "summaryColumns": [
           "MIN",
           "MAX",
           "SUM",
           "AVG"
       ]
   },
```

Schwellenwerte

Verwenden Sie dies, um Schwellenwerte auf Ihre Tabelle anzuwenden. Wenn ein Datenpunkt innerhalb eines Schwellenwerts liegt, wird seine Zelle mit der Schwellenwert-Farbe hervorgehoben. In der Quelle dargestellt als:

```
"annotations": {
    "horizontal": [
        {
            "label": string,
            "value": int,
            "fill": "above" | "below"
        }
    ]
```

```
}
```

Einheit in der Label-Spalte

Um anzuzeigen, welche Einheit mit der Metrik verknüpft ist, können Sie diese Option aktivieren, um die Einheit in der Label-Spalte neben der Bezeichnung anzuzeigen. In der Quelle dargestellt als:

```
"yAxis": {
    "left": {
        "showUnits": true | false
    }
}
```

Zeilen und Spalten invertieren

Dadurch wird die Tabelle so transformiert, dass die Datenpunkte von Spalten zu Zeilen wechseln und die Metriken zu Spalten werden. In der Quelle dargestellt als:

```
 "table": {
    "layout": "vertical" | "horizontal"
}
```

Fixierte Übersichtsspalten

Dadurch werden die Übersichtsspalten fixiert, sodass sie beim Scrollen sichtbar bleiben. Das Label ist bereits fixiert. In der Quelle dargestellt als:

```
"table": {
    "stickySummary": true | false
}
```

Nur Übersichtsspalten anzeigen

Dadurch wird verhindert, dass die Spalten von Datenpunkten angezeigt werden, sodass nur die Label- und Übersichtsspalten angezeigt werden. In der Quelle dargestellt als:

```
 "table": {
    "showTimeSeriesData": false | true
}
```

Live-Daten

Zeigt den neuesten Datenpunkt an, auch wenn er noch nicht vollständig aggregiert ist. In der Quelle dargestellt als:

```
"liveData": true | false
```

Format des Zahlen-Widgets

Zeigt vor dem Runden und Umrechnen so viele Ziffern an, wie in die Zelle passen. In der Quelle dargestellt als:

```
"singleValueFullPrecision": true | false
```

So fügen Sie ein Datentabellen-Widget zu einem Dashboard hinzu

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie die Schaltfläche +, wählen Sie Datentabelle und dann Weiter.
4. Suchen Sie auf der Registerkarte Durchsuchen nach den Metriken, die Sie im Tabellen-Widget anzeigen möchten. Wählen Sie dann die Metriken.
5. (Optional) Um das Layout der Tabelle zu ändern, wählen Sie die Registerkarte Optionen und dann Zeilen und Spalten umkehren.

   Sie können auch die Registerkarte Optionen verwenden, um zu ändern, welche Spalten in der Tabelle angezeigt werden, und um die in der Label-Spalte verwendete Einheit anzuzeigen.

   > **Tip**
   >
   > Um genauere Schwellenwerte anzuzeigen, wählen Sie Vor dem Runden so viele Ziffern wie möglich anzeigen.

6. (Optional) Um den Zeitbereich Ihres Datentabellen-Widgets zu ändern, wählen Sie einen der vordefinierten Zeitbereiche im oberen Bereich des Widgets. Die Zeitbereiche reichen von 1 Stunde bis 1 Woche. Um Ihren eigenen Zeitraum festzulegen, wählen Sie Custom (Benutzerdefiniert).

7. (Optional) Um den Zeitbereich Ihres Datentabellen-Widgets zu ändern, wählen Sie einen der vordefinierten Zeitbereiche im oberen Bereich des Widgets. Die Zeitbereiche reichen von 1 Stunde bis 1 Woche. Um Ihren eigenen Zeitraum festzulegen, wählen Sie Custom (Benutzerdefiniert).
8. (Optional) Damit dieses Widget weiterhin den von Ihnen ausgewählten Zeitbereich verwendet, auch wenn der Zeitbereich für den Rest des Dashboards später geändert wird, wählen Sie Zeitbereich beibehalten.
9. Wählen Sie Widget erstellen und dann Dashboard speichern.

So entfernen Sie ein Datentabellen-Widget aus einem Dashboard

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie in der oberen rechten Ecke des Widgets, das Sie entfernen möchten, Widget-Aktionen und dann Löschen.
4. Wählen Sie Save dashboard aus.

Diagramme auf einem CloudWatch Dashboard verknüpfen und deren Verknüpfung aufheben

Sie können die Diagramme auf dem Dashboard miteinander verknüpfen, damit andere Diagramme sich gleichzeitig mit vergrößern oder verkleinern, sobald Sie ein einzelnes Diagramm vergrößern oder verkleinern. Sie können die Verknüpfung mit einem Diagramm aufheben, um die Vergrößerung oder Verkleinerung auf ein Diagramm zu beschränken.

So verknüpfen Sie die Grafiken auf einem Dashboard

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie Actions (Aktionen) und dann Link graphics (Diagramme verknüpfen).

So heben Sie die Verknüpfung der Grafiken auf einem Dashboard auf

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.

3. Löschen Sie Actions (Aktionen) und dann Link graphics (Diagramme verknüpfen).

CloudWatch Dashboards teilen

Sie können Ihre CloudWatch Dashboards mit Personen teilen, die keinen direkten Zugriff auf Ihr AWS Konto haben. Auf diese Weise können Sie Dashboards für Teams, für Stakeholder und für Personen außerhalb Ihrer Organisation freigeben. Sie können Dashboards sogar auf großen Bildschirmen in Teambereichen anzeigen oder in Wikis und anderen Webseiten einbetten.

> **Warning**
>
> Alle Personen, für die Sie das Dashboard freigeben, erhalten die Berechtigungen, die unter Berechtigungen, die Personen erteilt werden, für die Sie das Dashboard freigeben für das Konto aufgelistet sind. Wenn Sie das Dashboard öffentlich freigeben, hat jeder, der über den Link zum Dashboard verfügt, diese Berechtigungen.
> Die `ec2:DescribeTags` Berechtigungen `cloudwatch:GetMetricData` und können nicht auf bestimmte Metriken oder EC2-Instances beschränkt werden, sodass die Personen mit Zugriff auf das Dashboard alle CloudWatch Metriken sowie die Namen und Tags aller EC2-Instances im Konto abfragen können.

Wenn Sie Dashboards freigeben, können Sie auf drei Arten festlegen, wer das Dashboard anzeigen kann:

- Teilen Sie ein einzelnes Dashboard und weisen Sie bis zu fünf E-Mail-Adressen von Personen zu, die das Dashboard einsehen können. Jeder dieser Benutzer erstellt sein eigenes Kennwort, das er eingeben muss, um das Dashboard anzuzeigen.
- Teilen Sie ein einzelnes Dashboard öffentlich, damit jeder, der über den Link verfügt, das Dashboard anzeigen kann.
- Teilen Sie alle CloudWatch Dashboards in Ihrem Konto und geben Sie einen Drittanbieter für Single Sign-On (SSO) für den Zugriff auf das Dashboard an. Alle Benutzer, die Mitglieder der Liste dieses SSO-Anbieters sind, können auf alle Dashboards im Konto zugreifen. Um dies zu ermöglichen, integrieren Sie den SSO-Anbieter in Amazon Cognito. Der SSO-Anbieter muss Security Assertion Markup Language (SAML) unterstützen. Weitere Informationen zu Amazon Cognito erhalten Sie unter Was ist Amazon Cognito?.

Für die gemeinsame Nutzung eines Dashboards fallen keine Gebühren an, aber für Widgets innerhalb eines gemeinsamen Dashboards fallen Gebühren zu Standardtarifen an. CloudWatch Weitere Informationen zur CloudWatch Preisgestaltung finden Sie unter CloudWatch Amazon-Preise.

Wenn Sie ein Dashboard teilen, werden Amazon Cognito Cognito-Ressourcen in der Region USA Ost (Nord-Virginia) erstellt.

> **Important**
>
> Ändern Sie keine Ressourcennamen und -identifikatoren, die durch den Prozess der Dashboardfreigabe erstellt wurden. Dazu gehören Amazon-Cognito- und IAM-Ressourcen. Das Ändern dieser Ressourcen kann zu unerwarteten und fehlerhaften Funktionen der gemeinsamen Dashboards führen.

> **Note**
>
> Wenn Sie ein Dashboard mit Metrik-Widgets mit Alarmanmerkungen freigeben, sehen die Personen, mit denen Sie das Dashboard freigeben, diese Widgets nicht. Stattdessen wird ein leeres Widget mit Text angezeigt, das besagt, dass das Widget nicht verfügbar ist. Sie werden weiterhin Metrik-Widgets mit Alarmanmerkungen sehen, wenn Sie das Dashboard selbst anzeigen.

Für die Freigabe eines Dashboards erforderliche Berechtigungen

Um Dashboards mit einer der folgenden Methoden freigeben zu können und um zu sehen, welche Dashboards bereits freigegeben wurden, müssen Sie als Benutzer oder mit einer IAM-Rolle angemeldet sein, die über bestimmte Berechtigungen verfügt.

Um Dashboards freigeben zu können, muss Ihr Benutzer oder Ihre IAM-Rolle über die Berechtigungen verfügen, die in der folgenden Richtlinie aufgeführt sind:

```
{
    "Effect": "Allow",
    "Action": [
        "iam:CreateRole",
        "iam:CreatePolicy",
```

```
        "iam:AttachRolePolicy",
        "iam:PassRole"
    ],
    "Resource": [
        "arn:aws:iam::*:role/service-role/CWDBSharing*",
        "arn:aws:iam::*:policy/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "cognito-idp:*",
        "cognito-identity:*",
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "cloudwatch:GetDashboard",
    ],
    "Resource": [
        "*"
        // or the ARNs of dashboards that you want to share
    ]
}
```

Um sehen zu können, welche Dashboards freigegeben sind, aber keine Dashboards freigeben können, kann ein Benutzer oder eine IAM-Rolle eine Richtlinie wie die folgende enthalten:

```
{
    "Effect": "Allow",
    "Action": [
        "cognito-idp:*",
        "cognito-identity:*"
    ],
    "Resource": [
        "*"
    ]
},
{
```

```
    "Effect": "Allow",
    "Action": [
        "cloudwatch:ListDashboards",
    ],
    "Resource": [
        "*"
    ]
}
```

Berechtigungen, die Personen erteilt werden, für die Sie das Dashboard freigeben

Wenn Sie ein Dashboard teilen, CloudWatch wird im Konto eine IAM-Rolle erstellt, die den Personen, mit denen Sie das Dashboard teilen, die folgenden Berechtigungen gewährt:

- `cloudwatch:GetInsightRuleReport`
- `cloudwatch:GetMetricData`
- `cloudwatch:DescribeAlarms`
- `ec2:DescribeTags`

> **Warning**
>
> Allen Personen, mit denen Sie das Dashboard teilen, werden diese Berechtigungen für das Konto erteilt. Wenn Sie das Dashboard öffentlich freigeben, hat jeder, der über den Link zum Dashboard verfügt, diese Berechtigungen.
> Die `ec2:DescribeTags` Berechtigungen `cloudwatch:GetMetricData` und können nicht auf bestimmte Metriken oder EC2-Instances beschränkt werden, sodass die Personen mit Zugriff auf das Dashboard alle CloudWatch Metriken sowie die Namen und Tags aller EC2-Instances im Konto abfragen können.

Wenn Sie ein Dashboard teilen, beschränken die CloudWatch erstellten Berechtigungen standardmäßig nur den Zugriff auf die Alarme und Contributor Insights-Regeln, die sich auf dem Dashboard befinden, wenn es geteilt wird. Wenn Sie dem Dashboard neue Alarme oder Contributor-Insights-Regeln hinzufügen und diese auch von den Personen angezeigt werden sollen, für die Sie das Dashboard freigegeben haben, müssen Sie die Richtlinie aktualisieren, um diese Ressourcen zuzulassen.

Ein einzelnes Dashboard für bestimmte Benutzer freigeben

Gehen Sie wie in diesem Abschnitt beschrieben vor, um ein Dashboard mit bis zu fünf E-Mail-Adressen Ihrer Wahl zu teilen.

> **Note**
>
> Standardmäßig sind alle CloudWatch Logs-Widgets auf dem Dashboard für Personen, mit denen Sie das Dashboard teilen, nicht sichtbar. Weitere Informationen finden Sie unter Zulassen von Benutzern, für die Sie freigeben, zum Anzeigen von Protokolltabellenwidgets. Standardmäßig sind zusammengesetzte Alarm-Widgets auf dem Dashboard für Personen, mit denen Sie das Dashboard teilen, nicht sichtbar. Weitere Informationen finden Sie unter Zulassen, dass Personen, mit denen Sie Inhalte teilen, zusammengesetzte Alarme sehen.

So geben Sie ein Dashboard für bestimmte Benutzer frei

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Dashboards aus.
3. Wählen Sie den Namen Ihres Dashboards aus.
4. Klicken Sie auf Aktionen,Freigeben von Dashboards.
5. Wählen Sie neben Dashboard freigeben und Benutzernamen und Passwort anfordern die Option Freigabe starten aus.
6. Geben Sie unter E-Mail-Adressen hinzufügen die E-Mail-Adressen ein, mit denen Sie das Dashboard teilen möchten. Sie können bis zu fünf E-Mail-Adressen angeben.
7. Wenn Sie alle E-Mail-Adressen eingegeben haben, lesen Sie die Vereinbarung und wählen Sie das Bestätigungsfeld aus. Wählen Sie dann Vorschaurichtlinie aus.
8. Bestätigen Sie, dass die freigegebenen Ressourcen Ihren Wünschen entsprechen, und wählen Sie Bestätigen und gemeinsam nutzbaren Link generieren aus.
9. Wählen Sie auf der nächsten Seite Link in die Zwischenablage kopieren aus. Sie können diesen Link dann in eine E-Mail einfügen und an die eingeladenen Benutzer senden. Sie erhalten automatisch eine separate E-Mail mit ihrem Benutzernamen und einem temporären Kennwort für die Verbindung mit dem Dashboard.

Ein einzelnes Dashboard öffentlich freigeben

Befolgen Sie die Schritte in diesem Abschnitt, um ein Dashboard öffentlich freizugeben. Dies kann nützlich sein, um das Dashboard auf einem großen Bildschirm in einem Teamraum anzuzeigen oder es in eine Wiki-Seite einzubetten.

> **Important**
>
> Das öffentliche Freigeben eines Dashboards macht es für jeden zugänglich, der den Link hat, ohne Authentifizierung. Führen Sie dies nur für Dashboards aus, die keine vertraulichen Informationen enthalten.

> **Note**
>
> Standardmäßig sind alle CloudWatch Logs-Widgets auf dem Dashboard für Personen, mit denen Sie das Dashboard teilen, nicht sichtbar. Weitere Informationen finden Sie unter Zulassen von Benutzern, für die Sie freigeben, zum Anzeigen von Protokolltabellenwidgets. Standardmäßig sind zusammengesetzte Alarm-Widgets auf dem Dashboard für Personen, mit denen Sie das Dashboard teilen, nicht sichtbar. Weitere Informationen finden Sie unter Zulassen, dass Personen, mit denen Sie Inhalte teilen, zusammengesetzte Alarme sehen.

So geben Sie ein Dashboard öffentlich frei

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Dashboards aus.
3. Wählen Sie den Namen Ihres Dashboards aus.
4. Klicken Sie auf Aktionen,Freigeben von Dashboards.
5. Wählen Sie neben Dashboard öffentlich freigeben die Option Freigabe starten aus.
6. Geben Sie **`Confirm`** in das Textfeld ein.
7. Lesen Sie die Vereinbarung und wählen Sie das Bestätigungsfeld aus. Wählen Sie dann Vorschaurichtlinie aus.
8. Bestätigen Sie, dass die freigegebenen Ressourcen Ihren Wünschen entsprechen, und wählen Sie Bestätigen und gemeinsam nutzbaren Link generieren aus.

9. Wählen Sie auf der nächsten Seite Link in die Zwischenablage kopieren aus. Sie können diesen Link dann freigeben. Jeder Benutzer, mit dem Sie den Link freigeben, kann ohne Anmeldeinformationen auf das Dashboard zugreifen.

Teilen Sie alle CloudWatch Dashboards im Konto mithilfe von SSO

Verwenden Sie die Schritte in diesem Abschnitt, um alle Dashboards in Ihrem Konto mithilfe von Single Sign-On (SSO) für Benutzer freizugeben.

> **Note**
>
> Standardmäßig sind alle CloudWatch Logs-Widgets im Dashboard für Personen, mit denen Sie das Dashboard teilen, nicht sichtbar. Weitere Informationen finden Sie unter Zulassen von Benutzern, für die Sie freigeben, zum Anzeigen von Protokolltabellenwidgets.
> Standardmäßig sind zusammengesetzte Alarm-Widgets auf dem Dashboard für Personen, mit denen Sie das Dashboard teilen, nicht sichtbar. Weitere Informationen finden Sie unter Zulassen, dass Personen, mit denen Sie Inhalte teilen, zusammengesetzte Alarme sehen.

Um Ihre CloudWatch Dashboards mit Benutzern zu teilen, die auf der Liste eines SSO-Anbieters stehen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Dashboards aus.
3. Wählen Sie den Namen Ihres Dashboards aus.
4. Klicken Sie auf Aktionen,Freigeben von Dashboards.
5. Wählen Sie Gehe zu CloudWatch Einstellungen.
6. Wenn der gewünschte SSO-Anbieter nicht unter Verfügbare SSO-Anbieter aufgeführt ist, wählen Sie SSO-Anbieter verwalten und befolgen Sie die Anweisungen in Richten Sie SSO für die gemeinsame Nutzung von CloudWatch Dashboards ein.

   Kehren Sie dann zur CloudWatch Konsole zurück und aktualisieren Sie den Browser. Der von Ihnen aktivierte SSO-Anbieter sollte nun in der Liste angezeigt werden.
7. Wählen Sie den gewünschten SSO-Anbieter in der Liste Verfügbare SSO-Anbieter aus.
8. Wählen Sie Änderungen speichern aus.

Richten Sie SSO für die gemeinsame Nutzung von CloudWatch Dashboards ein

Gehen Sie folgendermaßen vor, um die Dashboard-Freigabe über einen Drittanbieter für einmaliges Anmelden einzurichten, der SAML unterstützt.

> **Important**
>
> Es wird dringend empfohlen, Dashboards nicht mithilfe eines Nicht-SAML-Anbieters freizugeben. Dies führt zu dem Risiko, dass Drittpartien versehentlich den Zugriff auf die Dashboards Ihres Kontos gestatten.

So richten Sie einen SSO-Anbieter ein, um die Dashboard-Freigabe zu aktivieren

1. Integrieren Sie den SSO-Anbieter mit Amazon Cognito. Weitere Informationen finden Sie unter Integrieren von SAML-Identitätsanbietern von Drittanbietern in Amazon-Cognito-Benutzerpools.
2. Laden Sie die XML-Metadaten-Datei von Ihrem SSO-Anbieter herunter.
3. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
4. Wählen Sie im Navigationsbereich Settings (Einstellungen).
5. Wählen Sie im Abschnitt Dashboard-Freigabe die Option Konfigurieren aus.
6. Wählen Sie SSO-Anbieter verwalten aus.

   So öffnen Sie die Amazon-Cognito-Konsole in der Region USA Ost (Nord-Virginia) (us-east-1). Wenn keine Benutzerpools angezeigt werden, wurde die Amazon-Cognito-Konsole möglicherweise in einer anderen Region geöffnet. Ändern Sie in diesem Fall die Region in USA Ost (Nord-Virginia) us-east-1 und fahren Sie mit den nächsten Schritten fort.
7. Wählen Sie den CloudWatchDashboardSharingPool.
8. Wählen Sie im Navigationsbereich Identitätsanbieter.
9. Wählen Sie SAML.
10. Geben Sie unter Anbietername einen Namen für Ihren SSO-Anbieter ein.
11. Wählen Sie Datei auswählen und wählen Sie die Metadaten-XML-Datei aus, die Sie in Schritt 1 heruntergeladen haben.
12. Wählen Sie Create provider (Anbieter erstellen).

Sehen Sie, wie viele Ihrer Dashboards freigegeben sind

Sie können die CloudWatch Konsole verwenden, um zu sehen, wie viele Ihrer CloudWatch Dashboards derzeit mit anderen geteilt werden.

So sehen Sie, wie viele Ihrer Dashboards freigegeben werden

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Settings (Einstellungen).
3. Der Abschnitt Dashboard-Freigabe zeigt an, wie viele Dashboards freigegeben sind.
4. Um zu sehen, welche Dashboards freigegeben sind, wählen Sie unter Benutzername und Passwort und unter Öffentliche Dashboards die ***`Anzahl`*** freigegebener Dashboards aus.

Sehen Sie, welche Ihrer Dashboards freigegeben werden

Sie können die CloudWatch Konsole verwenden, um zu sehen, welche Ihrer Dashboards derzeit mit anderen geteilt werden.

So sehen Sie, welche Ihrer Dashboards freigegeben werden.

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Dashboards aus.
3. In der Liste der Dashboards finden Sie die Spalte Freigeben. Dashboards, für die das Symbol in dieser Spalte ausgefüllt ist, werden derzeit freigegeben.
4. Um zu sehen, mit welchen Benutzern ein Dashboard freigegeben wird, wählen Sie den Dashboardnamen und dann Aktionen, Dashboard freigeben aus.

   Die Seite Dashboard freigeben ***`Dashboard-Name`*** zeigt an, wie das Dashboard geteilt wird. Wenn Sie möchten, können Sie die Freigabe des Dashboards beenden, indem Sie Freigabe beenden auswählen.

So beenden Sie die Freigabe eines oder mehrerer Dashboards

Sie können die Freigabe eines einzelnen freigegebenen Dashboards beenden oder die Freigabe aller freigegebenen Dashboards auf einmal beenden.

So beenden Sie die Freigabe eines einzelnen Dashboards

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Dashboards aus.
3. Wählen Sie den Namen des freigegebenen Dashboards aus.
4. Klicken Sie auf Aktionen,Freigeben von Dashboards.
5. Wählen Sie Freigabe beenden.
6. Wählen Sie im Bestätigungsfeld Freigabe beenden aus.

So beenden Sie die Freigabe aller freigegebenen Dashboards

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Settings (Einstellungen).
3. Wählen Sie im Abschnitt Dashboard-Freigabe die Option Freigabe aller Dashboards beenden aus.
4. Wählen Sie im Bestätigungsfeld Freigabe aller Dashboards beenden aus.

Überprüfen der Berechtigungen für freigegebene Dashboards und Ändern des Berech

Führen Sie die Schritte in diesem Abschnitt aus, wenn Sie die Berechtigungen der Benutzer Ihrer freigegebenen Dashboards überprüfen oder den Umfang der freigegebenen Dashboard-Berechtigungen ändern möchten.

So überprüfen Sie freigegebene Dashboard-Berechtigungen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Dashboards aus.
3. Wählen Sie den Namen des freigegebenen Dashboards aus.
4. Klicken Sie auf Aktionen,Freigeben von Dashboards.
5. Wählen Sie unter Ressourcen die Option IAM-Rolle aus.
6. Wählen Sie in der IAM-Konsole die angezeigte Richtlinie aus.
7. (Optional) Um einzuschränken, welche Alarme Benutzer von freigegebenen Dashboards sehen können, wählen Sie Richtlinie bearbeiten und verschieben Sie die

`cloudwatch:DescribeAlarms`-Berechtigung von ihrer aktuellen Position in eine neue `Allow`-Anweisung, die nur die ARNs der Alarme auflistet, die von freigegebenen Dashboard-Benutzern angezeigt werden sollen. Sehen Sie sich das folgende -Beispiel an.

```
{
   "Effect": "Allow",
    "Action": "cloudwatch:DescribeAlarms",
    "Resource": [
        "AlarmARN1",
        "AlarmARN2"
    ]
}
```

Stellen Sie in diesem Fall sicher, dass Sie die `cloudwatch:DescribeAlarms`-Berechtigung aus einem Abschnitt der aktuellen Richtlinie entfernen, der wie folgt aussieht:

```
{
   "Effect": "Allow",
    "Action": [
        "cloudwatch:GetInsightRuleReport",
        "cloudwatch:GetMetricData",
        "cloudwatch:DescribeAlarms",
        "ec2:DescribeTags"
    ],
    "Resource": "*"
}
```

8. (Optional) Um den Umfang der Contributor-Insights-Regeln einzuschränken, die Benutzer des freigegebenen Dashboards sehen können, wählen Sie Richtlinie bearbeiten und verschieben Sie das `cloudwatch:GetInsightRuleReport` von seiner aktuellen Position in eine neue `Allow`-Anweisung, die nur die ARNs der Contributor-Insights-Regeln auflistet, die Sie sein möchten von freigegebenen Dashboard-Benutzern gesehen. Sehen Sie sich das folgende -Beispiel an.

```
{
   "Effect": "Allow",
    "Action": "cloudwatch:GetInsightRuleReport",
    "Resource": [
        "PublicContributorInsightsRuleARN1",
        "PublicContributorInsightsRuleARN2"
    ]
```

```
}
```

Stellen Sie in diesem Fall sicher, dass Sie die `cloudwatch:GetInsightRuleReport` aus einem Abschnitt der aktuellen Richtlinie entfernen, der wie folgt aussieht:

```
{
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetInsightRuleReport",
                "cloudwatch:GetMetricData",
                "cloudwatch:DescribeAlarms",
                "ec2:DescribeTags"
            ],
            "Resource": "*"
        }
```

Zulassen, dass Personen, mit denen Sie Inhalte teilen, zusammengesetzte Alarme sehen

Wenn Sie ein Dashboard freigeben, sind die zusammengesetzten Alarm-Widgets auf dem Dashboard standardmäßig nicht für die Personen sichtbar, für die Sie das Dashboard freigeben. Damit zusammengesetzte Alarm-Widgets sichtbar sind, müssen Sie der Dashboard-Freigaberichtlinie eine `DescribeAlarms: *`-Berechtigung hinzufügen. Diese Berechtigung würde wie folgt aussehen:

```
{
    "Effect": "Allow",
    "Action": "cloudwatch:DescribeAlarms",
    "Resource": "*"
}
```

Warning

Die vorhergehende Richtlinienanweisung gewährt Zugriff auf alle Alarme im Konto. Um den Gültigkeitsbereich von `cloudwatch:DescribeAlarms` zu reduzieren, müssen Sie eine `Deny`-Anweisung verwenden. Sie können der Richtlinie eine `Deny`-Anweisung hinzufügen und die ARNs der Alarme angeben, die Sie sperren möchten. Diese Zugriffsverweigerungs-Anweisung sollte folgendermaßen oder ähnlich aussehen:

```
{
    "Effect": "Allow",
    "Action": "cloudwatch:DescribeAlarms",
    "Resource": "*"
},
{
    "Effect": "Deny",
    "Action": "cloudwatch:DescribeAlarms",
    "Resource": [
        "SensitiveAlarm1ARN",
        "SensitiveAlarm1ARN"
    ]
}
```

Zulassen von Benutzern, für die Sie freigeben, zum Anzeigen von Protokolltabellenwidgets

Wenn Sie ein Dashboard teilen, sind die CloudWatch Logs Insights-Widgets, die sich auf dem Dashboard befinden, standardmäßig nicht für die Personen sichtbar, mit denen Sie das Dashboard teilen. Dies wirkt sich sowohl auf CloudWatch Logs Insights-Widgets aus, die jetzt existieren, als auch auf alle Widgets, die dem Dashboard hinzugefügt werden, nachdem Sie es geteilt haben.

Wenn Sie möchten, dass diese Personen CloudWatch Logs-Widgets sehen können, müssen Sie der IAM-Rolle Berechtigungen für die gemeinsame Nutzung von Dashboards hinzufügen.

Damit die Personen, mit denen Sie ein Dashboard teilen, die CloudWatch Logs-Widgets sehen können

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Dashboards aus.
3. Wählen Sie den Namen des freigegebenen Dashboards aus.
4. Klicken Sie auf Aktionen,Freigeben von Dashboards.
5. Wählen Sie unter Ressourcen die Option IAM-Rolle aus.
6. Wählen Sie in der IAM-Konsole die angezeigte Richtlinie aus.

7. Wählen Sie Richtlinie bearbeiten und fügen Sie die folgende Anweisung hinzu. Es wird empfohlen, in der neuen Anweisung nur die ARNs der Protokollgruppen anzugeben, die freigegeben werden sollen. Sehen Sie sich das folgende -Beispiel an.

```
{
            "Effect": "Allow",
            "Action": [
                "logs:FilterLogEvents",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:GetLogRecord",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "SharedLogGroup1ARN",
                "SharedLogGroup2ARN"
           ]
        },
```

8. Wählen Sie Save Changes.

Wenn Ihre IAM-Richtlinie für die Dashboard-Freigabe bereits diese fünf Berechtigungen mit * als Ressource enthält, empfehlen wir Ihnen dringend, die Richtlinie zu ändern und nur die ARNs der Protokollgruppen anzugeben, die freigegeben werden sollen. Wenn Ihr `Resource`-Abschnitt für diese Berechtigungen beispielsweise wie folgt lautet:

```
"Resource": "*"
```

Ändern Sie die Richtlinie, um nur die ARNs der Protokollgruppen anzugeben, die freigegeben werden sollen, wie im folgenden Beispiel gezeigt:

```
"Resource": [
  "SharedLogGroup1ARN",
  "SharedLogGroup2ARN"
]
```

Erlauben von Benutzern, für die Sie freigeben, dass benutzerdefinierte Widgets angezeigt werden

Wenn Sie ein Dashboard freigeben, sind die benutzerdefinierten Widgets, die sich auf dem Dashboard befinden, standardmäßig für die Personen, für die Sie das Dashboard freigeben, nicht sichtbar. Dies betrifft sowohl benutzerdefinierte Widgets, die jetzt vorhanden sind, als auch alle, die dem Dashboard hinzugefügt werden, nachdem Sie es freigegeben haben.

Wenn diese Personen benutzerdefinierte Widgets anzeigen können, müssen Sie der IAM-Rolle Berechtigungen für die Dashboard-Freigabe hinzufügen.

So lassen Sie den Personen, für die Sie ein Dashboard freigeben, die benutzerdefinierten Widgets anzeigen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Dashboards aus.
3. Wählen Sie den Namen des freigegebenen Dashboards aus.
4. Klicken Sie auf Aktionen,Freigeben von Dashboards.
5. Wählen Sie unter Ressourcen die Option IAM-Rolle aus.
6. Wählen Sie in der IAM-Konsole die angezeigte Richtlinie aus.
7. Wählen Sie Richtlinie bearbeiten und fügen Sie die folgende Anweisung hinzu. In der neuen Anweisung wird empfohlen, die ARNs nur der Lambda-Funktionen anzugeben, die freigegeben werden sollen. Sehen Sie sich das folgende -Beispiel an.

```
{
  "Sid": "Invoke",
  "Effect": "Allow",
  "Action": [
      "lambda:InvokeFunction"
  ],
  "Resource": [
    "LambdaFunction1ARN",
    "LambdaFunction2ARN"
  ]
 }
```

8. Wählen Sie Save Changes.

Wenn Ihre IAM-Richtlinie für die Dashboard-Freigabe diese Berechtigung bereits mit * als Ressource enthält, empfehlen wir Ihnen dringend, die Richtlinie zu ändern und nur die ARNs der Lambda-Funktionen anzugeben, die freigegeben werden sollen. Wenn Ihr `Resource`-Abschnitt für diese Berechtigungen beispielsweise wie folgt lautet:

```
"Resource": "*"
```

Ändern Sie die Richtlinie, um nur die ARNs der benutzerdefinierten Widgets anzugeben, die freigegeben werden sollen, wie im folgenden Beispiel gezeigt:

```
"Resource": [
  "LambdaFunction1ARN",
  "LambdaFunction2ARN"
]
```

Verwenden von Live-Daten

Sie können wählen, ob Ihre Metrik-Widgets Live-Daten anzeigen. Live-Daten sind Daten, die innerhalb der letzten Minute veröffentlicht und noch nicht vollständig aggregiert wurden.

- Wenn Live-Daten deaktiviert sind, werden nur Datenpunkte mit einem Aggregationszeitraum von mindestens einer Minute in der Vergangenheit angezeigt. Bei Verwendung von 5-Minuten-Zeiträumen wird der Datenpunkt für 12:35 von 12:35 zu 12:40 aggregiert und um 12:41 angezeigt.
- Wenn Live-Daten aktiviert sind, wird der neueste Datenpunkt angezeigt, sobald Daten im entsprechenden Aggregationsintervall veröffentlicht werden. Bei jeder Aktualisierung der Anzeige, ändert sich der aktuellste Datenpunkt möglicherweise, wenn neue Daten innerhalb dieses Aggregationszeitraums veröffentlicht werden. Wenn Sie eine kumulative Statistik wie Sum (Summe) oder Sample Count (Stichprobenanzahl) verwenden, kann die Verwendung von Live-Daten zu einem Abbruch am Ende des Diagramms führen.

Sie können Live-Daten für ein ganzes Dashboard oder für einzelne Widgets im Dashboard aktivieren.

So wählen Sie aus, ob Live-Daten auf Ihrem gesamten Dashboard verwendet werden sollen:

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.

3. Um Live-Daten für alle Widgets auf dem Dashbaord dauerhaft ein- oder auszuschalten, gehen Sie folgendermaßen vor:

   a. Wählen Sie Actions (Aktionen), Settings (Einstellungen), Bulk update live data (Massenaktualisierung von Live-Daten).

   b. Wählen Sie Live Data on (Live-Daten ein) oder „ Live Data off (Live-Daten aus) und dann Set (Einstellen)..

4. Um die Live-Dateneinstellungen jedes Widgets vorübergehend zu überschreiben, wählen Sie Aktionen. Führen Sie dann unter Überschreibungen neben Live-Daten einen der folgenden Schritte aus:

   - Wählen Sie On (An), um Live-Daten für alle Widgets vorübergehend zu aktivieren.
   - Wählen Sie Off (Aus), um Live-Daten für alle Widgets vorübergehend zu deaktivieren.
   - Wählen Sie Do not override (Nicht außer Kraft setzen), um die Live-Daten-Einstellung jedes Widgets beizubehalten.

So wählen Sie aus, ob Live-Daten für ein einzelnes Widget verwendet werden sollen:

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie ein Widget aus und dann Actions (Aktionen), Edit (Bearbeiten).
4. Wählen Sie die Registerkarte Graph options (Diagrammoptionen).
5. Aktivieren oder deaktivieren Sie das Kontrollkästchen unter Live Data (Live-Daten).

Anzeigen eines animierten Dashboards

Sie können sich ein animiertes Dashboard ansehen, das CloudWatch Metrikdaten wiedergibt, die im Laufe der Zeit erfasst wurden. Dies kann Ihnen helfen, Trends zu sehen, Präsentationen zu machen oder Probleme zu analysieren, nachdem sie auftreten.

Animierte Widgets im Dashboard umfassen Linien-Widgets, gestapelte Bereichs-Widgets, Zahlen-Widgets und Metrik-Explorer-Widgets. Kreisdiagramme, Balkendiagramme, Text-Widgets und Protokoll-Widgets werden im Dashboard angezeigt, aber nicht animiert.

So zeigen Sie ein animiertes Dashboard an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Dashboards aus.
3. Wählen Sie den Namen des Dashboards.
4. Wählen Sie Aktionen, Dashboard wiederholen.
5. (Optional) Wenn Sie die Animation starten, wird sie standardmäßig als Schiebefenster angezeigt. Wenn die Animation stattdessen als Animation angezeigt werden soll, wählen Sie das Lupensymbol, während die Animation angehalten ist, und setzen Sie den Zoom zurück. point-by-point
6. Um die Animation zu starten, wählen Sie die Schaltfläche „Wiedergabe“. Sie können auch die Zurück- und Vorwärtsschaltflächen auswählen, um zu anderen Zeitpunkten zu gelangen.
7. (Optional) Um das Zeitfenster für die Animation zu ändern, wählen Sie den Kalender und den Zeitraum aus.
8. Um die Geschwindigkeit der Animation zu ändern, wählen Sie Automatische Geschwindigkeit und wählen Sie die neue Geschwindigkeit.
9. Wenn Sie fertig sind, wählen Sie Animierung beenden.

Fügen Sie Ihrer CloudWatch Favoritenliste ein Dashboard hinzu

In der CloudWatch Konsole können Sie Dashboards, Alarme und Protokollgruppen zu einer Favoritenliste hinzufügen. Sie können über den Navigationsbereich auf die Favoritenliste zugreifen. Nachfolgend wird beschrieben, wie Sie ein Dashboard zur Favoritenliste hinzufügen.

So fügen Sie Ihrer Favoritenliste ein Dashboard hinzu

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Dashboards aus.
3. Wählen Sie aus der Liste der Dashboards das Sternsymbol neben dem Namen des Dashboards aus, das Sie bevorzugen möchten.

   - (Optional) Sie können ein Dashboard auch bevorzugen, indem Sie ein Dashboard aus der Liste auswählen und das Sternsymbol neben dem Dashboard-Namen auswählen.
4. Um auf die Favoritenliste zuzugreifen, wählen Sie Favorites and recents (Favoriten und kürzlich aufgerufene) im Navigationsbereich. Das Menü enthält zwei Spalten. Eines enthält

Ihre bevorzugten Dashboards, Alarme und Protokollgruppen, und die andere Spalte enthält die Dashboards, Alarme und Protokollgruppen, die Sie kürzlich besucht haben.

> **Tip**
>
> Im Navigationsbereich der CloudWatch Konsole können Sie Dashboards sowie Alarme und Protokollgruppen über das Menü „Favoriten“ und „Zuletzt verwendet“ zu Favoriten hinzufügen. Bewegen Sie in der Spalte Recently visited (Kürzlich aufgerufen) den Mauszeiger über das Dashboard, das Sie bevorzugen möchten, und wählen Sie das Sternsymbol daneben aus.

Ändern Sie die Einstellung für die Periodenüberschreibung oder das Aktualisierungsintervall für das Dashboard CloudWatch

Sie können angeben, wie die Zeitraumeinstellung von Diagrammen, die diesem Dashboard hinzugefügt wurden, beibehalten oder geändert werden.

Wenn ein automatischer Zeitraum oder ein persistenter Zeitbereich auf ein Widget angewendet wird, kann der gesamte Zeitbereich des Diagramms die von Ihnen festgelegten Zeiträume beeinflussen.

- Wenn der Zeitraum einen Tag oder weniger beträgt, werden die Zeitraumeinstellungen nicht geändert.
- Wenn der Zeitraum zwischen einem Tag und drei Tagen liegt, werden Zeiträume unter fünf Minuten auf 5 Minuten geändert.
- Wenn der Zeitraum mehr als drei Tage umfasst, werden Zeiträume, die unter einer Stunde liegen, auf eine Stunde geändert.

Die folgenden Schritte erklären, wie Sie mit der Konsole die Optionen zum Überschreiben von Zeiträumen ändern. Sie können sie auch ändern, indem Sie das `period0verride`-Feld in der JSON-Struktur des Dashboards verwenden. Weitere Informationen finden Sie unter Gesamtstruktur des Dashboard-Hauptteils.

So ändern Sie die Optionen für die Zeitraumüberschreibung

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

2. Wählen Sie Aktionen.
3. Wählen Sie unter Period override (Zeitraumüberschreibung) eine der folgenden Optionen aus:

   - Wählen Sie Auto (Automatisch) aus, damit der Zeitraum der Metriken auf jedem Diagramm automatisch an den Dashboard-Zeitraum angepasst wird.
   - Wählen Sie Do not override (Nicht überschreiben) aus, um sicherzustellen, dass die Zeitraumeinstellung des jeweiligen Diagramms immer berücksichtigt wird.
   - Wählen Sie eine der anderen Optionen aus, um zu bewirken, dass Diagramme, die dem Dashboard hinzugefügt wurden, immer die ausgewählte Zeit als Zeitraumeinstellung anpassen.

   Die Period override (Zeitraumüberschreibung) wird immer auf Auto (Automatisch) zurückgesetzt, wenn das Dashboard geschlossen oder der Browser aktualisiert wird. Abweichende Einstellungen für Period override (Zeitraumüberschreitung) können nicht gespeichert werden.

Sie können ändern, wie oft die Daten auf Ihrem CloudWatch Dashboard aktualisiert werden.

So ändern Sie das Aktualisierungsintervall für das Dashboard

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie im Menü Refresh options (Optionen aktualisieren) (in der oberen rechten Ecke) 10 Sekunden, 1 Minute, 2 Minuten, 5 Minutenoder 15 Minuten aus.

Ändern Sie den Zeitbereich oder das Zeitzonenformat eines CloudWatch Dashboards

Sie können den Zeitraum ändern, um Dashboard-Daten über Minuten, Stunden, Tage oder Wochen anzuzeigen. Sie können auch das Zeitzonenformat ändern, um Dashboard-Daten in UTC- oder lokaler Uhrzeit anzuzeigen. Lokale Uhrzeit ist die Zeitzone, die im Betriebssystem Ihres Computers angegeben ist.

> **Note**
>
> Wenn Sie ein Dashboard mit Diagrammen erstellen, die 100 oder mehr hochauflösende Metriken enthalten, empfehlen wir, den Zeitraum auf nicht länger als 1 Stunde festzulegen. Weitere Informationen finden Sie unter Hochauflösende Metriken.

> **Note**
>
> Wenn der Zeitraum eines Dashboards kürzer ist als der Zeitraum, der für ein Widget auf dem Dashboard verwendet wird, passiert Folgendes:
>
> - Das Widget wird so geändert, dass es die Datenmenge anzeigt, die einem vollständigen Zeitraum für dieses Widget entspricht, auch wenn dieser länger als der Dashboard-Zeitraum ist. Dadurch wird sichergestellt, dass es mindestens einen Datenpunkt im Diagramm gibt.
> - Die Startzeit des Zeitraums für diesen Datenpunkt wird nach hinten angepasst, um sicherzustellen, dass mindestens ein Datenpunkt angezeigt werden kann.

New console

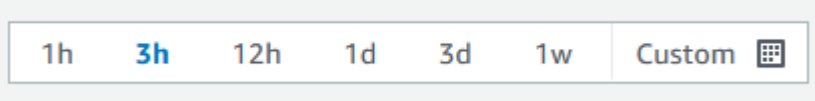
So ändern Sie den Dashboard-Zeitraum

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Führen Sie im Dashboard-Bildschirm einen der folgenden Schritte aus:
   - Wählen Sie im oberen Bereich des Dashboards einen der vordefinierten Zeitbereiche aus. Diese erstrecken sich von 1 Stunde bis 1 Woche (1 Std., 3 Std., 12 Std., 1 T. oder 1 W.).
   - Alternativ können Sie eine der folgenden benutzerdefinierten Zeitraum-Optionen wählen:
     - Wählen Sie das Menü Custom (Benutzerdefiniert) und dann die Registerkarte Relative (Relativ) aus. Wählen Sie einen Zeitraum von 1 Minute bis 15 Monaten aus.
     - Wählen Sie Custom (Benutzerdefiniert) und dann die Registerkarte Absolute (Absolut) aus. Verwenden Sie den Kalender oder die Textfelder, um Ihren Zeitraum festzulegen.

> **Tip**
>
> Wenn der Aggregationszeitraum auf Automatisch gesetzt ist, wenn Sie den Zeitraum eines Diagramms ändern, CloudWatch kann sich der Zeitraum ändern. Um den Zeitraum manuell festzulegen, wählen Sie das Dropdown-Menü Actions (Aktionen) und dann Period override (Zeitraum überschreiben) aus.

So ändern Sie das Dashboard-Zeitzonenformat

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie im oberen Bereich des Dashboards die Option Benutzerdefiniert.



4. Wählen Sie in der oberen rechten Ecke des angezeigten Felds UTC oder Local time (Ortszeit) aus der Dropdown-Liste aus.
5. Wählen Sie Apply (Anwenden) aus.

Old console

So ändern Sie den Dashboard-Zeitraum

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Führen Sie im Dashboard-Bildschirm einen der folgenden Schritte aus:
   - Wählen Sie im oberen Bereich des Dashboards einen der vordefinierten Zeitbereiche aus. Diese erstrecken sich von 1 Stunde bis 1 Woche (1 Std., 3 Std., 12 Std., 1 T., 3 T., oder 1 W.).
   - Alternativ können Sie eine der folgenden benutzerdefinierten Zeitraum-Optionen wählen:

- Wählen Sie das Dropdown-Menü Custom (Benutzerdefiniert) und dann Relative (Relativ) aus. Wählen Sie einen der vorgegebenen Zeiträume aus, die 1 Minute bis 15 Monate umfassen.
- Wählen Sie das Dropdown-Menü Custom (Benutzerdefiniert) und dann Absolute (Absolut) aus. Verwenden Sie den Kalender oder die Textfelder, um Ihren Zeitraum festzulegen.

> **Tip**
>
> Wenn der Aggregationszeitraum auf Automatisch gesetzt ist, wenn Sie den Zeitraum eines Diagramms ändern, CloudWatch kann sich der Zeitraum ändern. Um den Zeitraum manuell festzulegen, wählen Sie das Dropdown-Menü Actions (Aktionen) und dann Period override (Zeitraum überschreiben) aus.

So ändern Sie das Dashboard-Zeitzonenformat

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich die Option Dashboards und wählen Sie dann ein Dashboard.
3. Wählen Sie in der oberen rechten Ecke des Dashboard-Bildschirms die Dropdown-Liste Custom (Benutzerdefiniert).
4. Wählen Sie in der oberen rechten Ecke des angezeigten Feldes UTC oder Local timezone (Lokale Zeitzone) aus der Dropdown-Liste.

Verwenden Sie CloudWatch Amazon-Metriken

Metriken sind Daten über die Leistung Ihrer Systeme. Standardmäßig bieten viele Services kostenlose Metriken für Ressourcen (z. B. Amazon-EC2-Instances, Amazon-EBS-Volumes und Amazon-RDS-DB-Instances). Sie können auch eine detaillierte Überwachung für einige Ressourcen aktivieren, z. B. Ihre Amazon-EC2-Instances, oder Ihre eigenen Anwendungsmetriken veröffentlichen. Amazon CloudWatch kann alle Metriken in Ihrem Konto (sowohl AWS Ressourcen- als auch Anwendungsmetriken, die Sie angeben) für die Suche, grafische Darstellung und Alarme laden.

Metrikdaten werden 15 Monate lang aufbewahrt, sodass Sie sowohl up-to-the-minute Daten als auch historische Daten einsehen können.

Um Metriken in der Konsole grafisch darzustellen, können Sie CloudWatch Metrics Insights verwenden, eine leistungsstarke SQL-Abfrage-Engine, mit der Sie Trends und Muster in all Ihren Metriken in Echtzeit identifizieren können.

Inhalt

Grundlegende Überwachung und detaillierte Überwachung

CloudWatch bietet zwei Kategorien der Überwachung: Basisüberwachung und detaillierte Überwachung.

Viele AWS Dienste bieten eine grundlegende Überwachung, indem sie eine Reihe von Standardkennzahlen veröffentlichen, CloudWatch die für Kunden kostenlos sind. Wenn Sie eine dieser Optionen verwenden AWS-Services, ist die grundlegende Überwachung standardmäßig automatisch aktiviert. Eine Liste der Services, die grundlegende Überwachung bieten, finden Sie unter AWS Dienste, die CloudWatch Metriken veröffentlichen.

Eine detaillierte Überwachung wird nur von einigen Services angeboten. Es fallen auch Gebühren an. Um es für einen AWS Dienst zu verwenden, müssen Sie es aktivieren. Weitere Informationen zur Preisgestaltung finden Sie unter CloudWatch Amazon-Preise.

Detaillierte Überwachungsoptionen unterscheiden sich je nach den Services, die sie anbieten. Beispielsweise bietet die detaillierte Überwachung von Amazon EC2 häufigere Metriken, die in Intervallen von einer Minute veröffentlicht werden, statt den Fünf-Minuten-Intervallen, die in der Amazon-EC2-Basisüberwachung verwendet werden. Eine detaillierte Überwachung von Amazon S3 und Amazon Managed Streaming for Apache Kafka bedeutet detailliertere Metriken.

In verschiedenen AWS Diensten hat die detaillierte Überwachung auch unterschiedliche Namen. In Amazon EC2 wird es beispielsweise als detaillierte Überwachung, in AWS Elastic Beanstalk Amazon S3 als erweiterte Überwachung und in Amazon S3 als Anforderungsmetriken bezeichnet.

Durch die Verwendung einer detaillierten Überwachung für Amazon EC2 können Sie Ihre Amazon-EC2-Ressourcen besser verwalten, damit Sie Trends finden und schneller Maßnahmen ergreifen können. Für Amazon S3 sind Anforderungsmetriken in Abständen von einer Minute verfügbar, damit Sie operative Probleme schnell erkennen und umgehend handeln können. Wenn Sie auf Amazon MSK die Überwachung auf den Ebenen `PER_BROKER`, `PER_TOPIC_PER_BROKER` oder `PER_TOPIC_PER_PARTITIO`Naktivieren, erhalten Sie zusätzliche Metriken, die mehr Sichtbarkeit bieten.

In der folgenden Tabelle sind die Services aufgelistet, die detaillierte Überwachung bieten. Sie enthält auch Links zur Dokumentation für diese Services, die mehr über die detaillierte Überwachung erklären und Anweisungen zur Aktivierung enthalten.

Service	Dokumentation
Amazon API Gateway	Abmessungen für API-Gateway-Metriken

Service	Dokumentation
Amazon CloudFront	Zusätzliche CloudFront Vertriebsmetriken anzeigen
Amazon EC2	Aktivieren oder Deaktivieren der detaillierten Überwachung für Ihre Instances
Elastic Beanstalk	Erweiterte Zustandsberichte und Überwachung
Amazon-Kinesis-Data-Streams	Erweiterte Metriken auf Shard-Ebene
Amazon MSK	Amazon MSK-Metriken für die Überwachung mit CloudWatch
Amazon S3	Amazon S3 S3-Anforderungsmetriken in CloudWatch

Service	Dokumentation
Amazon SES	Sammeln Sie CloudWatch detaillierte Monitoring-Metriken mithilfe von Amazon SES Event Publishing.

Darüber hinaus CloudWatch bietet es out-of-the-box Überwachungslösungen mit detaillierteren Metriken und vorab erstellten Dashboards für einige AWS Dienste, wie in der folgenden Tabelle dargestellt.

Service	Dokumentation der Funktionen
Lambda	Lambda-Einblicke
Amazon ECS	Container-Einblicke für Amazon ECS
Amazon EKS	Container Insights für Amazon EKS und Kubernetes

Fragen Sie Ihre Metriken mit Metrics Insights ab CloudWatch

CloudWatch Metrics Insights ist eine leistungsstarke SQL-Abfrage-Engine, mit der Sie Ihre Metriken maßstabsgetreu abfragen können. Sie können Trends und Muster in all Ihren CloudWatch Metriken in Echtzeit identifizieren.

Sie können auch Alarme für alle Metrics-Insights-Abfragen einrichten, die eine einzelne Zeitreihe zurückgeben. Dies kann besonders nützlich sein, um Alarme zu erstellen, die aggregierte Metriken einer Flotte Ihrer Infrastruktur oder Ihrer Anwendungen überwachen. Wenn Sie den Alarm einmal erstellen, passt er sich dynamisch an, wenn Ressourcen zur Flotte hinzugefügt oder aus ihr entfernt werden.

Sie können eine CloudWatch Metrics Insights-Abfrage in der Konsole mit dem CloudWatch Metrics Insights-Abfrage-Editor ausführen. Sie können eine CloudWatch Metrics Insights-Abfrage auch mit dem AWS CLI oder einem AWS SDK ausführen, indem Sie `GetMetricData` oder ausführen`PutDashboard`. Für Abfragen, die Sie mit dem CloudWatch Metrics Insights-Abfrage-Editor ausführen, fallen keine Gebühren an. Weitere Informationen zur CloudWatch Preisgestaltung finden Sie unter CloudWatch Amazon-Preise.

Mit dem Abfrage-Editor von CloudWatch Metrics Insights können Sie aus einer Vielzahl von vorgefertigten Beispielabfragen wählen und auch Ihre eigenen Abfragen erstellen. Während Sie Ihre Abfragen erstellen, können Sie eine Builder-Ansicht verwenden, um Ihre vorhandenen Metriken und Dimensionen zu durchsuchen. Alternativ können Sie eine Editor-Ansicht verwenden, um Abfragen manuell zu schreiben.

Sie können auch natürliche Sprache verwenden, um CloudWatch Metrics Insights-Abfragen zu erstellen. Stellen Sie dazu Fragen zu den Daten, nach denen Sie suchen, oder beschreiben Sie sie. Diese KI-gestützte Funktion generiert auf der Grundlage Ihrer Aufforderung eine Abfrage und line-by-line erklärt, wie die Abfrage funktioniert. Weitere Informationen finden Sie unter Verwenden natürlicher Sprache zum Generieren und Aktualisieren von CloudWatch Metrics Insights-Abfragen.

Mit Metrics Insights können Sie Abfragen im großen Maßstab ausführen. Durch die Verwendung der Klausel GROUP BY können Sie Ihre Metriken flexibel in Echtzeit in separate Zeitreihen pro bestimmten Dimensionswert gruppieren. Da Metrics-Insights-Abfragen eine ORDER BY-Funktion beinhalten, können Sie Metrics Insights verwenden, um Abfragen vom Typ „Top N“ zu erstellen. Abfragen vom Typ „Top N“ können beispielsweise Millionen von Metriken in Ihrem Konto scannen und die 10 Instances zurückgeben, die am meisten CPU verbrauchen. Dies kann Ihnen helfen, Latenzprobleme in Ihren Anwendungen zu lokalisieren und zu beheben.

Themen

- Verwenden Sie natürliche Sprache, um CloudWatch Metrics Insights-Abfragen zu generieren und zu aktualisieren
- SQL-Inferenz
- Beispielabfragen für Metriken Insights
- Limits für Metric Insights
- Glossar zu Metric Insights
- Problembehebung bei Metrics Insights

Erstellen Ihrer Abfragen

Sie können eine CloudWatch Metrics Insights-Abfrage mit der CloudWatch Konsole AWS CLI, den oder den AWS SDKs ausführen. In der Konsole ausgeführte Abfragen sind gebührenfrei. Weitere Informationen zur CloudWatch Preisgestaltung finden Sie unter CloudWatch Amazon-Preise.

Weitere Informationen zur Verwendung der AWS SDKs zur Durchführung einer Metrics Insights-Abfrage finden Sie unter GetMetricData.

Gehen Sie folgendermaßen vor, um eine Abfrage mit der CloudWatch Konsole auszuführen:

Ihre Metriken mit Metrics Insights abfragen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken) All metrics (Alle Metriken) aus.
3. Wählen Sie die Registerkarte Queries (Abfragen) aus.
4. (Optional) Um eine vorgefertigte Beispielabfrage auszuführen, wählen Sie Add query (Abfrage hinzufügen) und wählen Sie die auszuführende Abfrage aus. Wenn Sie mit dieser Abfrage zufrieden sind, können Sie den Rest dieses Verfahrens überspringen. Oder Sie können den Editor auswählen, um die Beispielabfrage zu bearbeiten. Wählen Sie anschließend Run (Ausführen) aus, um die geänderte Abfrage auszuführen.
5. Um eine eigene Abfrage zu erstellen, können Sie entweder die Ansicht Builder oder Editor verwenden oder auch eine Kombination aus beiden. Sie können jederzeit zwischen den beiden Ansichten wechseln und Ihre laufende Arbeit in beiden Ansichten anzeigen.

   In der Ansicht Builder können Sie den Metrik-Namespace, den Metriknamen, den Filter, die Gruppe und die Bestelloptionen durchsuchen und auswählen. Für jede dieser Optionen bietet

Ihnen der Abfrage-Generator eine Liste von Auswahlmöglichkeiten in Ihrer Umgebung zur Auswahl.

In der Ansicht Editor können Sie mit dem Schreiben Ihrer Abfrage beginnen. Während der Eingabe bietet der Editor Vorschläge basierend auf den Zeichen, die Sie bisher eingegeben haben.

6. Wenn Sie mit Ihrer Abfrage zufrieden sind, klicken Sie auf Run (Ausführen).
7. (Optional) Eine andere Möglichkeit, eine abgebildete Abfrage zu bearbeiten, besteht darin, die Registerkarte Graphed metrics (Grafisch dargestellte Metriken) auszuwählen und anschließend das Bearbeitungssymbol neben der Abfrageformel in der Spalte Details auszuwählen.
8. (Optional) Um eine Abfrage aus dem Diagramm zu entfernen, wählen Sie Graphed metrics (Grafisch dargestellte Metriken) und anschließend das Symbol X auf der rechten Seite der Zeile, in der Ihre Abfrage angezeigt wird.

Abfragekomponenten und Syntax von Metrics Insights

CloudWatch Die Syntax von Metrics Insights lautet wie folgt.

```
SELECT FUNCTION(metricName)
FROM namespace | SCHEMA(...)
[ WHERE labelKey OPERATOR labelValue [AND ... ] ]
[ GROUP BY labelKey [ , ... ] ]
[ ORDER BY FUNCTION() [ DESC | ASC ] ]
[ LIMIT number ]
```

Die möglichen Klauseln in einer Metrics-Insights-Abfrage lauten wie folgt. Bei keinem der Schlüsselwörter wird Groß- und Kleinschreibung berücksichtigt, aber bei den Bezeichnern wie den Namen von Metriken, Namespaces und Dimensionen wird Groß- und Kleinschreibung beachtet.

SELECT

Erforderlich Gibt die Funktion an, die zum Aggregieren von Beobachtungen in jedem Zeit-Bucket verwendet werden soll (durch den angegebenen Zeitraum festgelegt). Gibt auch den Namen der abzufragenden Metrik an.

Die gültigen Werte für FUNCTION sind `AVG`, `COUNT`, `MAX`, `MIN` und `SUM`.

- `AAVG` berechnet den Durchschnitt der Beobachtungen, die mit der Abfrage übereinstimmen.

- `COUNT` gibt die Anzahl der Beobachtungen zurück, die mit der Abfrage übereinstimmen.
- `MAX` gibt den Maximalwert der Beobachtungen zurück, die mit der Abfrage übereinstimmen.
- `MIN` gibt den Minimalwert der Beobachtungen zurück, die mit der Abfrage übereinstimmen.
- `SUM` berechnet die Summe der Beobachtungen, die mit der Abfrage übereinstimmen.

FROM

Erforderlich Gibt die Quelle der Metrik an. Sie können entweder den Metrik-Namespace angeben, der die abzufragende Metrik enthält, oder eine SCHEMA-Tabellenfunktion. Beispiele für Metrik-Namespaces sind `"AWS/EC2"`, `"AWS/Lambda"` und Metrik-Namespaces, die Sie für Ihre benutzerdefinierten Metriken erstellt haben.

Metrik-Namespaces, die / enthalten oder jedes andere Zeichen, das kein Buchstabe, keine Zahl und kein Unterstrich ist, müssen von doppelten Anführungszeichen umgeben sein. Weitere Informationen finden Sie unter Was braucht Anführungszeichen oder Escape-Zeichen?.

SCHEMA

Eine optionale Tabellenfunktion, die innerhalb einer FROM-Klausel verwendet werden kann. Verwenden Sie SCHEMA, um die Abfrageergebnisse nur auf die Metriken zu reduzieren, die genau mit einer Liste von Dimensionen übereinstimmen, oder auf Metriken, die keine Dimensionen haben.

Wenn Sie eine SCHEMA-Klausel verwenden, muss diese mindestens ein Argument enthalten. Das erste Argument muss der Metrik-Namespace sein, der abgefragt wird. Wenn Sie SCHEMA nur mit diesem Namespace-Argument angeben, werden die Ergebnisse nur auf Metriken beschränkt, die keine Dimensionen haben.

Wenn Sie SCHEMA mit zusätzlichen Argumenten angeben, müssen die zusätzlichen Argumente nach dem Namespace-Argument Bezeichnungsschlüssel sein. Bezeichnungsschlüssel müssen Dimensionsnamen sein. Wenn Sie einen oder mehrere dieser Bezeichnungsschlüssel angeben, werden die Ergebnisse nur auf die Metriken beschränkt, die genau diesen Dimensionssatz aufweisen. Die Reihenfolge dieser Bezeichnungsschlüssel spielt dabei keine Rolle.

Zum Beispiel:

- SELECT AVG(CPUUtilization) FROM "AWS/EC2" entspricht allen `CPUUtilization`-Metriken im `AWS/EC2`-Namespace, unabhängig von ihren Dimensionen, und gibt eine einzelne aggregierte Zeitreihe zurück.

- SELECT AVG(CPUUtilization) FROM SCHEMA("AWS/EC2") entspricht nur den `CPUUtilization`-Metriken im `AWS/EC2`-Namespace, für die keine Dimensionen definiert sind.
- SELECT AVG (CPUUtilization) FROM SCHEMA („AWS/EC2", InstanceId) entspricht nur den `CPUUtilization` Metriken, für die berichtet wurde, CloudWatch mit genau einer Dimension,. `InstanceId`
- SELECT SUM (RequestCount) FROM SCHEMA („aws/ApplicationELB“ LoadBalancer, AvailabilityZone) entspricht nur den `RequestCount` Metriken, an die CloudWatch von gemeldet wurde, `AWS/ApplicationELB` mit genau zwei Dimensionen, und. `LoadBalancer` `AvailabilityZone`

WHERE

Optional. Filtert die Ergebnisse nur nach den Metriken, die mit dem angegebenen Ausdruck übereinstimmen, wobei bestimmte Bezeichnungswerte für einen oder mehrere Bezeichnungsschlüssel verwendet werden. Beispielsweise filtert WHERE InstanceType = 'c3.4xlarge' die Ergebnisse nur **`c3.4xlarge`** nach Instance-Typen und WHERE! InstanceType = 'c3.4xlarge' filtert die Ergebnisse nach allen Instanztypen außer. `c3.4xlarge`

Wenn Sie eine Abfrage in einem Überwachungskonto ausführen, können Sie `WHERE AWS.AccountId` verwenden, um die Ergebnisse auf das von Ihnen angegebene Konto zu beschränken. `WHERE AWS.AccountId=444455556666` fragt beispielsweise nur Metriken von Konto `444455556666` ab. Um Ihre Abfrage nur auf Metriken im Überwachungskonto selbst zu beschränken, verwenden Sie `WHERE AWS.AccountId=CURRENT_ACCOUNT_ID()`.

Bezeichnungswerte müssen immer mit einfachen Anführungszeichen eingeschlossen sein.

Unterstützte Operatoren

Die WHERE-Klausel unterstützt die folgenden Operatoren:

- = Der Bezeichnungswert muss mit der angegebenen Zeichenfolge übereinstimmen.
- != Der Bezeichnungswert muss nicht mit der angegebenen Zeichenfolge übereinstimmen.
- AND Beide angegebenen Bedingungen müssen zutreffend sein, damit sie übereinstimmen. Sie können mehrere AND-Schlüsselwörter verwenden, um zwei oder mehr Bedingungen anzugeben.

GROUP BY

Optional. Gruppiert die Abfrageergebnisse in mehrere Zeitreihen, die jeweils einem anderen Wert für den angegebenen Bezeichnungsschlüssel oder die angegebenen Schlüssel entsprechen.

Beispielsweise gibt `GROUP BY InstanceId` für jeden Wert eine andere Zeitreihe für jeden Wert von `InstanceId` zurück. Bei Verwendung von `GROUP BY ServiceName, Operation` wird eine andere Zeitreihe für jede mögliche Kombination der Werte von `ServiceName` und `Operation` erstellt.

Mit der Klausel GROUP BY werden die Ergebnisse standardmäßig in alphabetischer aufsteigender Reihenfolge angeordnet, wobei die in der Klausel GROUP BY angegebenen Bezeichnungssequenzen verwendet werden. Fügen Sie die Klausel ORDER BY zu Ihrer Anfrage hinzu, um die Reihenfolge der Ergebnisse zu ändern.

Wenn Sie eine Abfrage in einem Überwachungskonto ausführen, können Sie `GROUP BY AWS.AccountId` verwenden, um die Ergebnisse anhand der Konten zu gruppieren, von denen sie stammen.

> **Note**
>
> Wenn einige der übereinstimmenden Metriken keinen bestimmten Bezeichnungsschlüssel enthalten, der in der Klausel GROUP BY angegeben ist, wird eine Nullgruppe mit dem Namen `Other` zurückgegeben. Wenn Sie beispielsweise `GROUP BY ServiceName, Operation` angeben und einige der zurückgegebenen Metriken `ServiceName` nicht als Dimension enthalten, werden diese Metriken so angezeigt, als hätten sie `Other` als Wert für `ServiceName`.

ORDER BY

Optional. Gibt die Reihenfolge an, die für die zurückgegebene Zeitreihe verwendet werden soll, wenn die Abfrage mehr als eine Zeitreihe zurückgibt. Die Reihenfolge basiert auf den Werten, die von der FUNCTION gefunden werden, die Sie in der Klausel ORDER BY angegeben haben. Die FUNCTION wird verwendet, um einen einzelnen Skalarwert aus jeder zurückgegebenen Zeitreihe zu berechnen. Dieser Wert wird verwendet, um die Reihenfolge zu bestimmen.

Sie geben auch an, ob aufsteigend (ASC) oder absteigend (DESC) verwendet werden soll. Wenn Sie dies auslassen, ist die Standardeinstellung aufsteigend (ASC).

Fügen Sie zum Beispiel die Klausel `ORDER BY MAX() DESC` hinzu, werden die Ergebnisse nach dem maximalen Datenpunkt in absteigender Reihenfolge sortiert, der innerhalb des Zeitraums beobachtet wird. Das bedeutet, dass die Zeitreihe mit dem höchsten maximalen Datenpunkt zuerst zurückgegeben wird.

Die gültigen Funktionen, die innerhalb der Klausel ORDER BY verwendet werden, lauten `AVG()`, `COUNT()`, `MAX()`, `MIN()` und `SUM()`.

Wenn Sie die Klausel ORDER BY mit der Klausel LIMIT verwenden, ist die resultierende Abfrage eine „Top N“-Abfrage. ORDER BY ist auch nützlich für Abfragen, die möglicherweise eine große Anzahl von Metriken zurückgeben, da keine Abfrage mehr als 500 Zeitreihen zurückgeben kann. Wenn eine Abfrage mehr als 500 Zeitreihen entspricht und Sie die Klausel ORDER BY verwenden, werden die Zeitreihen sortiert und die 500 Zeitreihen, die in der Sortierreihenfolge zuerst stehen, werden zurückgegeben.

LIMIT

Optional. Beschränkt die Anzahl der von der Abfrage zurückgegebenen Zeitreihen auf den von Ihnen angegebenen Wert. Der Maximalwert, den Sie angeben können, ist 500, und eine Abfrage, die kein LIMIT angibt, kann ebenfalls maximal 500 Zeitreihen zurückgeben.

Wenn Sie die Klausel LIMIT mit der Klausel ORDER BY verwenden, erhalten Sie eine „Top N“-Abfrage.

Was braucht Anführungszeichen oder Escape-Zeichen?

In einer Abfrage müssen Bezeichnungswerte immer mit einfachen Anführungszeichen eingeschlossen sein. Zum Beispiel SELECT MAX (CPUUtilization) FROM „AWS/EC2" WHERE = ". AutoScalingGroupName my-production-fleet

Metrik-Namespaces, Metriknamen und Bezeichnungsschlüssel, die andere Zeichen als Buchstaben, Zahlen und Unterstriche (_) enthalten, müssen mit doppelten Anführungszeichen eingeschlossen sein. Beispiel: SELECT MAX("My.Metric").

Wenn eine dieser Abfragen ein doppeltes Anführungszeichen oder ein einzelnes Anführungszeichen enthält (z. B. `Bytes"Input"`), muss jedem Anführungszeichen ein umgekehrter Schrägstrich vorangestellt werden, siehe SELECT AVG("Bytes\"Input\"").

Wenn ein Metrik-Namespace, ein Metrikname oder ein Bezeichnungsschlüssel ein Wort enthält, das ein reserviertes Schlüsselwort in Metrics Insights ist, muss dieses auch in doppelten Anführungszeichen eingeschlossen sein. Wenn Sie beispielsweise eine Metrik mit dem Namen `LIMIT` haben, würden Sie `SELECT AVG("LIMIT")` benutzen. Es ist auch möglich, einen Namespace, einen Metriknamen oder eine Bezeichnung in doppelte Anführungszeichen einzuschließen, auch wenn kein reserviertes Schlüsselwort enthalten ist.

Eine vollständige Liste der reservierten Wörter finden Sie unter Reservierte Schlüsselwörter.

Eine umfangreiche Abfrage erstellen, Schritt für Schritt

Dieser Abschnitt veranschaulicht das Erstellen eines vollständigen Beispiels, das Schritt für Schritt alle möglichen Klauseln verwendet.

Wir beginnen mit der folgenden Abfrage, die alle `RequestCount`-Metriken im Application Load Balancer aggregiert, die mit den beiden Dimensionen `LoadBalancer` und `AvailabilityZone` gesammelt werden.

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
```

Wenn wir nun Metriken von einem bestimmten Load Balancer sehen möchten, können wir die Klausel WHERE hinzufügen, um die Metriken zu beschränken, die nur für die Metriken zurückgegeben werden, bei denen der Wert der `LoadBalancer`-Dimension `app/load-balancer-1` ist.

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
```

Die vorhergehende Abfrage aggregiert die `RequestCount`-Metriken aus allen Availability Zones für diesen Load Balancer in eine Zeitreihe. Wenn wir verschiedene Zeitreihen für jede Availability Zone sehen möchten, können wir die Klausel GROUP BY hinzufügen.

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
GROUP BY AvailabilityZone
```

Als Nächstes möchten wir diese Ergebnisse möglicherweise so anordnen, dass zuerst die höchsten Werte angezeigt werden. Die folgende ORDER BY-Klausel ordnet die Zeitreihe in absteigender Reihenfolge um den Maximalwert an, den jede Zeitreihe während des Abfragezeitraums gemeldet hat:

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
```

```
WHERE LoadBalancer = 'app/load-balancer-1'
GROUP BY AvailabilityZone
ORDER BY MAX() DESC
```

Wenn wir in erster Linie an einer „Top N“-Abfrage interessiert sind, können wir eine LIMIT-Klausel verwenden. Dieses letzte Beispiel beschränkt die Ergebnisse auf die Zeitreihe mit den fünf höchsten MAX-Werten.

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
GROUP BY AvailabilityZone
ORDER BY MAX() DESC
LIMIT 5
```

Beispiele kontoübergreifender Abfragen

Diese Beispiele sind gültig, wenn sie in einem Konto ausgeführt werden, das als Überwachungskonto für kontenübergreifende Observability eingerichtet wurde. CloudWatch

Im folgenden Beispiel werden alle Amazon-EC2-Instances im Quellkonto 123456789012 durchsucht und der Durchschnitt zurückgegeben.

```
SELECT AVG(CpuUtilization)
FROM "AWS/EC2"
WHERE AWS.AccountId ='123456789012'
```

Im folgenden Beispiel wird die `CPUUtilization`-Metrik in `AWS/EC2` in allen verknüpften Quellkonten abgefragt und die Ergebnisse nach Konto-ID und Instance-Typ gruppiert.

```
SELECT AVG(CpuUtilization)
FROM "AWS/EC2"
GROUP BY AWS.AccountId, InstanceType
```

Im folgenden Beispiel wird die `CPUUtilization` im Überwachungskonto selbst abgefragt.

```
SELECT AVG(CpuUtilization)
FROM "AWS/EC2"
WHERE AWS.AccountId = CURRENT_ACCOUNT_ID()
```

Reservierte Schlüsselwörter

Bei den folgenden Schlüsselwörtern handelt es sich um reservierte Schlüsselwörter in CloudWatch Metrics Insights. Wenn sich eines dieser Wörter in einem Namespace, einem Metriknamen oder einem Bezeichnungsschlüssel in einer Abfrage befindet, müssen Sie es in doppelte Anführungszeichen einschließen. Bei reservierten Schlüsselwörtern wird Groß- und Kleinschreibung nicht berücksichtigt.

```
"ABORT" "ABORTSESSION" "ABS" "ABSOLUTE" "ACCESS" "ACCESSIBLE" "ACCESS_LOCK" "ACCOUNT"
 "ACOS" "ACOSH" "ACTION" "ADD" "ADD_MONTHS"
"ADMIN" "AFTER" "AGGREGATE" "ALIAS" "ALL" "ALLOCATE" "ALLOW" "ALTER" "ALTERAND" "AMP"
 "ANALYSE" "ANALYZE" "AND" "ANSIDATE" "ANY" "ARE" "ARRAY",
"ARRAY_AGG" "ARRAY_EXISTS" "ARRAY_MAX_CARDINALITY" "AS" "ASC" "ASENSITIVE" "ASIN"
 "ASINH" "ASSERTION" "ASSOCIATE" "ASUTIME" "ASYMMETRIC" "AT",
"ATAN" "ATAN2" "ATANH" "ATOMIC" "AUDIT" "AUTHORIZATION" "AUX" "AUXILIARY" "AVE"
 "AVERAGE" "AVG" "BACKUP" "BEFORE" "BEGIN" "BEGIN_FRAME" "BEGIN_PARTITION",
"BETWEEN" "BIGINT" "BINARY" "BIT" "BLOB" "BOOLEAN" "BOTH" "BREADTH" "BREAK" "BROWSE"
 "BT" "BUFFERPOOL" "BULK" "BUT" "BY" "BYTE" "BYTEINT" "BYTES" "CALL",
"CALLED" "CAPTURE" "CARDINALITY" "CASCADE" "CASCADED" "CASE" "CASESPECIFIC" "CASE_N"
 "CAST" "CATALOG" "CCSID" "CD" "CEIL" "CEILING" "CHANGE" "CHAR",
"CHAR2HEXINT" "CHARACTER" "CHARACTERS" "CHARACTER_LENGTH" "CHARS" "CHAR_LENGTH" "CHECK"
 "CHECKPOINT" "CLASS" "CLASSIFIER" "CLOB" "CLONE" "CLOSE" "CLUSTER",
"CLUSTERED" "CM" "COALESCE" "COLLATE" "COLLATION" "COLLECT" "COLLECTION" "COLLID"
 "COLUMN" "COLUMN_VALUE" "COMMENT" "COMMIT" "COMPLETION" "COMPRESS" "COMPUTE",
"CONCAT" "CONCURRENTLY" "CONDITION" "CONNECT" "CONNECTION" "CONSTRAINT" "CONSTRAINTS"
 "CONSTRUCTOR" "CONTAINS" "CONTAINSTABLE" "CONTENT" "CONTINUE" "CONVERT",
"CONVERT_TABLE_HEADER" "COPY" "CORR" "CORRESPONDING" "COS" "COSH" "COUNT" "COVAR_POP"
 "COVAR_SAMP" "CREATE" "CROSS" "CS" "CSUM" "CT" "CUBE" "CUME_DIST",
"CURRENT" "CURRENT_CATALOG" "CURRENT_DATE" "CURRENT_DEFAULT_TRANSFORM_GROUP"
 "CURRENT_LC_CTYPE" "CURRENT_PATH" "CURRENT_ROLE" "CURRENT_ROW" "CURRENT_SCHEMA",
"CURRENT_SERVER" "CURRENT_TIME" "CURRENT_TIMESTAMP" "CURRENT_TIMEZONE"
 "CURRENT_TRANSFORM_GROUP_FOR_TYPE" "CURRENT_USER" "CURRVAL" "CURSOR" "CV" "CYCLE"
 "DATA",
"DATABASE" "DATABASES" "DATABLOCKSIZE" "DATE" "DATEFORM" "DAY" "DAYS" "DAY_HOUR"
 "DAY_MICROSECOND" "DAY_MINUTE" "DAY_SECOND" "DBCC" "DBINFO" "DEALLOCATE" "DEC",
"DECFLOAT" "DECIMAL" "DECLARE" "DEFAULT" "DEFERRABLE" "DEFERRED" "DEFINE" "DEGREES"
 "DEL" "DELAYED" "DELETE" "DENSE_RANK" "DENY" "DEPTH" "DEREF" "DESC" "DESCRIBE",
"DESCRIPTOR" "DESTROY" "DESTRUCTOR" "DETERMINISTIC" "DIAGNOSTIC" "DIAGNOSTICS"
 "DICTIONARY" "DISABLE" "DISABLED" "DISALLOW" "DISCONNECT" "DISK" "DISTINCT",
"DISTINCTROW" "DISTRIBUTED" "DIV" "DO" "DOCUMENT" "DOMAIN" "DOUBLE" "DROP" "DSSIZE"
 "DUAL" "DUMP" "DYNAMIC" "EACH" "ECHO" "EDITPROC" "ELEMENT" "ELSE" "ELSEIF",
"EMPTY" "ENABLED" "ENCLOSED" "ENCODING" "ENCRYPTION" "END" "END-EXEC" "ENDING"
 "END_FRAME" "END_PARTITION" "EQ" "EQUALS" "ERASE" "ERRLVL" "ERROR" "ERRORFILES",
```

```
"ERRORTABLES" "ESCAPE" "ESCAPED" "ET" "EVERY" "EXCEPT" "EXCEPTION" "EXCLUSIVE" "EXEC"
 "EXECUTE" "EXISTS" "EXIT" "EXP" "EXPLAIN" "EXTERNAL" "EXTRACT" "FALLBACK
"FALSE" "FASTEXPORT" "FENCED" "FETCH" "FIELDPROC" "FILE" "FILLFACTOR" "FILTER" "FINAL"
 "FIRST" "FIRST_VALUE" "FLOAT" "FLOAT4" "FLOAT8" "FLOOR"
"FOR" "FORCE" "FOREIGN" "FORMAT" "FOUND" "FRAME_ROW" "FREE" "FREESPACE" "FREETEXT"
 "FREETEXTTABLE" "FREEZE" "FROM" "FULL" "FULLTEXT" "FUNCTION"
"FUSION" "GE" "GENERAL" "GENERATED" "GET" "GIVE" "GLOBAL" "GO" "GOTO" "GRANT" "GRAPHIC"
 "GROUP" "GROUPING" "GROUPS" "GT" "HANDLER" "HASH"
"HASHAMP" "HASHBAKAMP" "HASHBUCKET" "HASHROW" "HAVING" "HELP" "HIGH_PRIORITY" "HOLD"
 "HOLDLOCK" "HOUR" "HOURS" "HOUR_MICROSECOND" "HOUR_MINUTE"
"HOUR_SECOND" "IDENTIFIED" "IDENTITY" "IDENTITYCOL" "IDENTITY_INSERT" "IF" "IGNORE"
 "ILIKE" "IMMEDIATE" "IN" "INCLUSIVE" "INCONSISTENT" "INCREMENT"
"INDEX" "INDICATOR" "INFILE" "INHERIT" "INITIAL" "INITIALIZE" "INITIALLY" "INITIATE"
 "INNER" "INOUT" "INPUT" "INS" "INSENSITIVE" "INSERT" "INSTEAD"
"INT" "INT1" "INT2" "INT3" "INT4" "INT8" "INTEGER" "INTEGERDATE" "INTERSECT"
 "INTERSECTION" "INTERVAL" "INTO" "IO_AFTER_GTIDS" "IO_BEFORE_GTIDS"
"IS" "ISNULL" "ISOBID" "ISOLATION" "ITERATE" "JAR" "JOIN" "JOURNAL" "JSON_ARRAY"
 "JSON_ARRAYAGG" "JSON_EXISTS" "JSON_OBJECT" "JSON_OBJECTAGG"
"JSON_QUERY" "JSON_TABLE" "JSON_TABLE_PRIMITIVE" "JSON_VALUE" "KEEP" "KEY" "KEYS"
 "KILL" "KURTOSIS" "LABEL" "LAG" "LANGUAGE" "LARGE" "LAST"
"LAST_VALUE" "LATERAL" "LC_CTYPE" "LE" "LEAD" "LEADING" "LEAVE" "LEFT" "LESS" "LEVEL"
 "LIKE" "LIKE_REGEX" "LIMIT" "LINEAR" "LINENO" "LINES"
"LISTAGG" "LN" "LOAD" "LOADING" "LOCAL" "LOCALE" "LOCALTIME" "LOCALTIMESTAMP" "LOCATOR"
 "LOCATORS" "LOCK" "LOCKING" "LOCKMAX" "LOCKSIZE" "LOG"
"LOG10" "LOGGING" "LOGON" "LONG" "LONGBLOB" "LONGTEXT" "LOOP" "LOWER" "LOW_PRIORITY"
 "LT" "MACRO" "MAINTAINED" "MAP" "MASTER_BIND"
"MASTER_SSL_VERIFY_SERVER_CERT" "MATCH" "MATCHES" "MATCH_NUMBER" "MATCH_RECOGNIZE"
 "MATERIALIZED" "MAVG" "MAX" "MAXEXTENTS" "MAXIMUM" "MAXVALUE"
"MCHARACTERS" "MDIFF" "MEDIUMBLOB" "MEDIUMINT" "MEDIUMTEXT" "MEMBER" "MERGE" "METHOD"
 "MICROSECOND" "MICROSECONDS" "MIDDLEINT" "MIN" "MINDEX"
"MINIMUM" "MINUS" "MINUTE" "MINUTES" "MINUTE_MICROSECOND" "MINUTE_SECOND" "MLINREG"
 "MLOAD" "MLSLABEL" "MOD" "MODE" "MODIFIES" "MODIFY"
"MODULE" "MONITOR" "MONRESOURCE" "MONSESSION" "MONTH" "MONTHS" "MSUBSTR" "MSUM"
 "MULTISET" "NAMED" "NAMES" "NATIONAL" "NATURAL" "NCHAR" "NCLOB"
"NE" "NESTED_TABLE_ID" "NEW" "NEW_TABLE" "NEXT" "NEXTVAL" "NO" "NOAUDIT" "NOCHECK"
 "NOCOMPRESS" "NONCLUSTERED" "NONE" "NORMALIZE" "NOT" "NOTNULL"
"NOWAIT" "NO_WRITE_TO_BINLOG" "NTH_VALUE" "NTILE" "NULL" "NULLIF" "NULLIFZERO" "NULLS"
 "NUMBER" "NUMERIC" "NUMPARTS" "OBID" "OBJECT" "OBJECTS"
"OCCURRENCES_REGEX" "OCTET_LENGTH" "OF" "OFF" "OFFLINE" "OFFSET" "OFFSETS" "OLD"
 "OLD_TABLE" "OMIT" "ON" "ONE" "ONLINE" "ONLY" "OPEN" "OPENDATASOURCE"
"OPENQUERY" "OPENROWSET" "OPENXML" "OPERATION" "OPTIMIZATION" "OPTIMIZE"
 "OPTIMIZER_COSTS" "OPTION" "OPTIONALLY" "OR" "ORDER" "ORDINALITY" "ORGANIZATION"
"OUT" "OUTER" "OUTFILE" "OUTPUT" "OVER" "OVERLAPS" "OVERLAY" "OVERRIDE" "PACKAGE" "PAD"
 "PADDED" "PARAMETER" "PARAMETERS" "PART" "PARTIAL" "PARTITION"
```

```
"PARTITIONED" "PARTITIONING" "PASSWORD" "PATH" "PATTERN" "PCTFREE" "PER" "PERCENT"
 "PERCENTILE" "PERCENTILE_CONT" "PERCENTILE_DISC" "PERCENT_RANK" "PERIOD" "PERM"
"PERMANENT" "PIECESIZE" "PIVOT" "PLACING" "PLAN" "PORTION" "POSITION" "POSITION_REGEX"
 "POSTFIX" "POWER" "PRECEDES" "PRECISION" "PREFIX" "PREORDER"
"PREPARE" "PRESERVE" "PREVVAL" "PRIMARY" "PRINT" "PRIOR" "PRIQTY" "PRIVATE"
 "PRIVILEGES" "PROC" "PROCEDURE" "PROFILE" "PROGRAM" "PROPORTIONAL"
"PROTECTION" "PSID" "PTF" "PUBLIC" "PURGE" "QUALIFIED" "QUALIFY" "QUANTILE" "QUERY"
 "QUERYNO" "RADIANS" "RAISERROR" "RANDOM" "RANGE" "RANGE_N" "RANK"
"RAW" "READ" "READS" "READTEXT" "READ_WRITE" "REAL" "RECONFIGURE" "RECURSIVE" "REF"
 "REFERENCES" "REFERENCING" "REFRESH" "REGEXP" "REGR_AVGX" "REGR_AVGY"
"REGR_COUNT" "REGR_INTERCEPT" "REGR_R2" "REGR_SLOPE" "REGR_SXX" "REGR_SXY" "REGR_SYY"
 "RELATIVE" "RELEASE" "RENAME" "REPEAT" "REPLACE" "REPLICATION"
"REPOVERRIDE" "REQUEST" "REQUIRE" "RESIGNAL" "RESOURCE" "RESTART" "RESTORE" "RESTRICT"
 "RESULT" "RESULT_SET_LOCATOR" "RESUME" "RET" "RETRIEVE" "RETURN"
"RETURNING" "RETURNS" "REVALIDATE" "REVERT" "REVOKE" "RIGHT" "RIGHTS" "RLIKE" "ROLE"
 "ROLLBACK" "ROLLFORWARD" "ROLLUP" "ROUND_CEILING" "ROUND_DOWN"
"ROUND_FLOOR" "ROUND_HALF_DOWN" "ROUND_HALF_EVEN" "ROUND_HALF_UP" "ROUND_UP" "ROUTINE"
 "ROW" "ROWCOUNT" "ROWGUIDCOL" "ROWID" "ROWNUM" "ROWS" "ROWSET"
"ROW_NUMBER" "RULE" "RUN" "RUNNING" "SAMPLE" "SAMPLEID" "SAVE" "SAVEPOINT" "SCHEMA"
 "SCHEMAS" "SCOPE" "SCRATCHPAD" "SCROLL" "SEARCH" "SECOND" "SECONDS"
"SECOND_MICROSECOND" "SECQTY" "SECTION" "SECURITY" "SECURITYAUDIT" "SEEK" "SEL"
 "SELECT" "SEMANTICKEYPHRASETABLE" "SEMANTICSIMILARITYDETAILSTABLE"
"SEMANTICSIMILARITYTABLE" "SENSITIVE" "SEPARATOR" "SEQUENCE" "SESSION" "SESSION_USER"
 "SET" "SETRESRATE" "SETS" "SETSESSRATE" "SETUSER" "SHARE" "SHOW"
"SHUTDOWN" "SIGNAL" "SIMILAR" "SIMPLE" "SIN" "SINH" "SIZE" "SKEW" "SKIP" "SMALLINT"
 "SOME" "SOUNDEX" "SOURCE" "SPACE" "SPATIAL" "SPECIFIC" "SPECIFICTYPE"
"SPOOL" "SQL" "SQLEXCEPTION" "SQLSTATE" "SQLTEXT" "SQLWARNING" "SQL_BIG_RESULT"
 "SQL_CALC_FOUND_ROWS" "SQL_SMALL_RESULT" "SQRT" "SS" "SSL" "STANDARD"
"START" "STARTING" "STARTUP" "STAT" "STATE" "STATEMENT" "STATIC" "STATISTICS" "STAY"
 "STDDEV_POP" "STDDEV_SAMP" "STEPINFO" "STOGROUP" "STORED" "STORES"
"STRAIGHT_JOIN" "STRING_CS" "STRUCTURE" "STYLE" "SUBMULTISET" "SUBSCRIBER" "SUBSET"
 "SUBSTR" "SUBSTRING" "SUBSTRING_REGEX" "SUCCEEDS" "SUCCESSFUL"
"SUM" "SUMMARY" "SUSPEND" "SYMMETRIC" "SYNONYM" "SYSDATE" "SYSTEM" "SYSTEM_TIME"
 "SYSTEM_USER" "SYSTIMESTAMP" "TABLE" "TABLESAMPLE" "TABLESPACE" "TAN"
"TANH" "TBL_CS" "TEMPORARY" "TERMINATE" "TERMINATED" "TEXTSIZE" "THAN" "THEN"
 "THRESHOLD" "TIME" "TIMESTAMP" "TIMEZONE_HOUR" "TIMEZONE_MINUTE" "TINYBLOB"
"TINYINT" "TINYTEXT" "TITLE" "TO" "TOP" "TRACE" "TRAILING" "TRAN" "TRANSACTION"
 "TRANSLATE" "TRANSLATE_CHK" "TRANSLATE_REGEX" "TRANSLATION" "TREAT"
"TRIGGER" "TRIM" "TRIM_ARRAY" "TRUE" "TRUNCATE" "TRY_CONVERT" "TSEQUAL" "TYPE" "UC"
 "UESCAPE" "UID" "UNDEFINED" "UNDER" "UNDO" "UNION" "UNIQUE"
"UNKNOWN" "UNLOCK" "UNNEST" "UNPIVOT" "UNSIGNED" "UNTIL" "UPD" "UPDATE" "UPDATETEXT"
 "UPPER" "UPPERCASE" "USAGE" "USE" "USER" "USING" "UTC_DATE"
"UTC_TIME" "UTC_TIMESTAMP" "VALIDATE" "VALIDPROC" "VALUE" "VALUES" "VALUE_OF"
 "VARBINARY" "VARBYTE" "VARCHAR" "VARCHAR2" "VARCHARACTER" "VARGRAPHIC"
```

```
"VARIABLE" "VARIADIC" "VARIANT" "VARYING" "VAR_POP" "VAR_SAMP" "VCAT" "VERBOSE"
 "VERSIONING" "VIEW" "VIRTUAL" "VOLATILE" "VOLUMES" "WAIT" "WAITFOR"
"WHEN" "WHENEVER" "WHERE" "WHILE" "WIDTH_BUCKET" "WINDOW" "WITH" "WITHIN"
 "WITHIN_GROUP" "WITHOUT" "WLM" "WORK" "WRITE" "WRITETEXT" "XMLCAST" "XMLEXISTS"
"XMLNAMESPACES" "XOR" "YEAR" "YEARS" "YEAR_MONTH" "ZEROFILL" "ZEROIFNULL" "ZONE"
```

Alarme für Metrics-Insights-Abfragen erstellen

Sie können Alarme in Metrics-Insights-Abfragen erstellen. Dies hilft Ihnen dabei, Alarme zu haben, die mehrere Ressourcen verfolgen, ohne dass sie später aktualisiert werden müssen. Die Abfrage erfasst neue Ressourcen und Ressourcen, die sich ändern. Sie können beispielsweise einen Alarm erstellen, der die CPU-Auslastung Ihrer Flotte überwacht, und der Alarm bewertet automatisch neue Instances, die Sie nach der Erstellung des Alarms starten.

In einem Monitoring-Konto, das für CloudWatch kontoübergreifende Beobachtbarkeit eingerichtet ist, können Ihre Metrics Insights-Alarme Ressourcen in Quellkonten und im Monitoring-Konto selbst überwachen. Weitere Informationen darüber, wie Sie Ihre Alarmanfragen auf ein bestimmtes Konto beschränken oder die Ergebnisse nach Konto-ID gruppieren können, finden Sie in den Abschnitten `WHERE` und `GROUP BY` unter Abfragekomponenten und Syntax von Metrics Insights.

Inhalt

Einen Metrics-Insights-Alarm erstellen

So erstellen Sie einen Alarm für eine Metrics-Insights-Abfrage mit der Konsole

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken) All metrics (Alle Metriken) aus.
3. Wählen Sie die Registerkarte Queries (Abfragen) aus.
4. (Optional) Um eine vorgefertigte Beispielabfrage auszuführen, wählen Sie Add query (Abfrage hinzufügen) und wählen Sie die auszuführende Abfrage aus. Oder Sie können den Editor auswählen, um die Beispielabfrage zu bearbeiten. Wählen Sie anschließend Run (Ausführen) aus, um die geänderte Abfrage auszuführen.

5. Um eine eigene Abfrage zu erstellen, können Sie entweder die Ansicht Builder oder Editor verwenden oder auch eine Kombination aus beiden. Sie können jederzeit zwischen den beiden Ansichten wechseln und Ihre laufende Arbeit in beiden Ansichten anzeigen.

   In der Ansicht Builder können Sie den Metrik-Namespace, den Metriknamen, den Filter, die Gruppe und die Bestelloptionen durchsuchen und auswählen. Für jede dieser Optionen bietet Ihnen der Abfrage-Generator eine Liste von Auswahlmöglichkeiten in Ihrer Umgebung zur Auswahl.

   In der Ansicht Editor können Sie mit dem Schreiben Ihrer Abfrage beginnen. Während der Eingabe bietet der Editor Vorschläge basierend auf den Zeichen, die Sie bisher eingegeben haben.

   > **Important**
   >
   > Um einen Alarm für eine Metrics-Insights-Abfrage auszulösen, muss die Abfrage eine einzelne Zeitreihe zurückgeben. Wenn sie eine GROUP-BY-Anweisung enthält, muss die GROUP-BY-Anweisung in einen metrischen mathematischen Ausdruck eingebettet werden, der nur eine Zeitreihe als Endergebnis des Ausdrucks zurückgibt.

6. Wenn Sie mit Ihrer Abfrage zufrieden sind, klicken Sie auf Run (Ausführen).
7. Wählen Sie Alarm erstellen aus.
8. Geben Sie unter Conditions (Bedingungen) Folgendes an:

   a. Geben Sie für Wann immer die ***Metrik*** ist an, ob die Metrik größer, kleiner oder gleich dem Schwellenwert sein muss. Geben Sie unter than... (dann ...) den Schwellenwert an.

   b. Wählen Sie Additional configuration (Zusätzliche Konfiguration). Geben Sie unter Datapoints to alarm (Datenpunkte für Alarm) an, wie viele Auswertungszeiträume (Datenpunkte) im Status `ALARM` sein müssen, damit der Alarm ausgelöst wird. Wenn die beiden Werte hier übereinstimmen, erstellen Sie einen Alarm, der in den Status `ALARM` wechselt, wenn entsprechend viele aufeinanderfolgende Zeiträume überschritten werden.

      Um einen M aus N Alarm zu erstellen, geben Sie eine niedrigere Zahl für den ersten Wert als für den zweiten Wert an. Weitere Informationen finden Sie unter Auswerten eines Alarms.

c. Wählen Sie für Missing data treatment (Behandlung von fehlenden Daten) aus, wie sich der Alarm verhalten soll, wenn einige Datenpunkte fehlen. Weitere Informationen finden Sie unter Konfiguration, wie Alarme fehlende Daten behandeln CloudWatch .

9. Wählen Sie Weiter.
10. Wählen Sie unter Notification (Benachrichtigung) ein SNS-Thema aus, das benachrichtigt werden soll, wenn sich der Alarm im Status `ALARM`, `OK` oder `INSUFFICIENT_DATA` befindet.

    Um zu erreichen, dass der Alarm mehrere Benachrichtigungen für den gleichen Alarmstatus oder für verschiedene Statuswerte sendet, wählen Sie Benachrichtigung hinzufügen.

    Damit der Alarm keine Benachrichtigungen sendet, wählen Sie Remove (Entfernen).
11. Um den Alarm Auto-Scaling-, EC2- oder Systems-Manager-Aktionen durchführen zu lassen, wählen Sie die entsprechende Schaltfläche und wählen Sie den Alarmstatus und die auszuführende Aktion. Alarme können Aktionen des Systems Manager nur ausführen, wenn sie in den ALARM-Zustand wechseln. Weitere Informationen zu Systems Manager Manager-Aktionen finden Sie unter Konfiguration für CloudWatch die Erstellung OpsItems aus Alarmen und Incident-Erstellung.

    > **Note**
    >
    > Um einen Alarm zu erstellen, der eine SSM-Incident-Manager-Aktion ausführt, müssen Sie über bestimmte Berechtigungen verfügen. Weitere Informationen finden Sie unter Beispiele für identitätsbasierte Richtlinien für AWS Systems Manager Incident Manager.
12. Wenn Sie fertig sind, wählen Sie Weiter.
13. Geben Sie einen Namen und eine Beschreibung für den Alarm ein. Der Name darf nur ASCII-Zeichen enthalten. Wählen Sie anschließend Weiter.
14. Bestätigen Sie unter Preview and create (Vorschau und erstellen), dass die Informationen und Bedingungen den Anforderungen entsprechen, und wählen Sie dann Create alarm (Alarm erstellen).

Um einen Alarm für eine Metrics Insights-Abfrage zu erstellen, verwenden Sie AWS CLI

- Verwenden Sie den `put-metric-alarm`-Befehl und geben Sie im `metrics`-Parameter eine Metrics-Insights-Abfrage an. Mit dem folgenden Befehl wird beispielsweise ein Alarm ausgelöst,

der in den Zustand ALARM wechselt, wenn die CPU-Auslastung einer Ihrer Instances über 50 % steigt.

```
aws cloudwatch put-metric-alarm --alarm-name Metrics-Insights-alarm --
evaluation-periods 1 --comparison-operator GreaterThanThreshold --metrics
 '[{"Id":"m1","Expression":"SELECT MAX(CPUUtilization) FROM SCHEMA(\"AWS/EC2\",
 InstanceId)", "Period":60}]' --threshold 50
```

Fälle mit Teildaten

Wenn die für den Alarm verwendete Metrics-Insights-Abfrage mehr als 10 000 Metriken entspricht, wird der Alarm anhand der ersten 10 000 Metriken ausgewertet, die die Abfrage findet. Das bedeutet, dass der Alarm anhand von Teildaten ausgewertet wird.

Sie können die folgenden Methoden verwenden, um herauszufinden, ob ein Metrics-Insights-Alarm seinen Alarmstatus derzeit anhand von Teildaten auswertet:

- Wenn Sie in der Konsole einen Alarm auswählen, um die Seite Details aufzurufen, erscheint auf dieser Seite die Meldung Evaluation warning: Not evaluating all data (Bewertungswarnung: Es werden nicht alle Daten bewertet).
- Sie sehen den Wert `PARTIAL_DATA` in dem `EvaluationState` Feld, wenn Sie den AWS CLI Befehl describe-alarms oder die DescribeAlarmsAPI verwenden.

Alarme veröffentlichen auch Ereignisse an Amazon, EventBridge wenn es in den Status „Teildaten“ übergeht, sodass Sie eine EventBridge Regel erstellen können, um auf diese Ereignisse zu achten. In diesen Ereignissen hat das `evaluationState`-Feld den Wert `PARTIAL_DATA`. Im Folgenden wird ein Beispiel gezeigt.

```
{
    "version": "0",
    "id": "12345678-3bf9-6a09-dc46-12345EXAMPLE",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-11-08T11:26:05Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:my-alarm-name"
    ],
```

```
    "detail": {
        "alarmName": "my-alarm-name",
        "state": {
            "value": "ALARM",
            "reason": "Threshold Crossed: 3 out of the last 3 datapoints [20000.0
 (08/11/22 11:25:00), 20000.0 (08/11/22 11:24:00), 20000.0 (08/11/22 11:23:00)] were
 greater than the threshold (0.0) (minimum 1 datapoint for OK -> ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2022-11-08T11:26:05.399+0000\",\"startDate\":\"2022-11-08T11:23:00.000+0000\",
\"period\":60,\"recentDatapoints\":[20000.0,20000.0,20000.0],\"threshold\":0.0,
\"evaluatedDatapoints\":[{\"timestamp\":\"2022-11-08T11:25:00.000+0000\",\"value
\":20000.0}]}",
            "timestamp": "2022-11-08T11:26:05.401+0000",
            "evaluationState": "PARTIAL_DATA"
        },
        "previousState": {
            "value": "INSUFFICIENT_DATA",
            "reason": "Unchecked: Initial alarm creation",
            "timestamp": "2022-11-08T11:25:51.227+0000"
        },
        "configuration": {
            "metrics": [
                {
                    "id": "m2",
                    "expression": "SELECT SUM(PartialDataTestMetric) FROM
 partial_data_test",
                    "returnData": true,
                    "period": 60
                }
            ]
        }
    }
}
```

Wenn die Abfrage für den Alarm eine GROUP-BY-Anweisung enthält, die anfänglich mehr als 500 Zeitreihen zurückgibt, wird der Alarm anhand der ersten 500 Zeitreihen ausgewertet, die die Abfrage findet. Wenn Sie jedoch eine ORDER-BY-Klausel verwenden, werden alle von der Abfrage zu verarbeitenden Zeitreihen sortiert, und die 500, die nach der ORDER-BY-Klausel den höchsten oder niedrigsten Wert haben, werden verwendet, um den Alarm zu bewerten.

Metrics-Insights-Abfragen mit Metrikberechnungen verwenden

Sie können eine Metrics-Insights-Abfrage als Eingabe für eine metrische mathematische Funktion verwenden. Weitere Informationen zu Metrikberechnungen finden Sie unter Verwenden von Metrikberechnungen.

Eine Metrics-Insights-Abfrage, die keine GROUP BY-Klausel enthält, gibt eine einzige Zeitreihe zurück. Daher können die zurückgegebenen Ergebnisse mit jeder metrischen mathematischen Funktion verwendet werden, die eine einzelne Zeitreihe als Eingabe verwendet.

Eine Metrics-Insights-Abfrage, die eine GROUP BY-Klausel enthält, gibt mehrere Zeitreihen zurück. Daher können die zurückgegebenen Ergebnisse mit jeder metrischen mathematischen Funktion verwendet werden, die mehrere Zeitreihen als Eingabe verwendet.

Die folgende Abfrage gibt beispielsweise die Gesamtzahl der für jeden Bucket in der Region heruntergeladenen Bytes als Array von Zeitreihen zurück:

```
SELECT SUM(BytesDownloaded)
FROM SCHEMA("AWS/S3", BucketName, FilterId)
WHERE FilterId = 'EntireBucket'
GROUP BY BucketName
```

In einem Diagramm in der Konsole oder in einer GetMetricDataOperation werden q1 die Ergebnisse dieser Abfrage angezeigt. Diese Abfrage gibt das Ergebnis in Byte zurück. Wenn Sie also das Ergebnis stattdessen als MB sehen möchten, können Sie die folgende mathematische Funktion verwenden:

```
q1/1024/1024
```

Verwenden Sie natürliche Sprache, um CloudWatch Metrics Insights-Abfragen zu generieren und zu aktualisieren

Diese Funktion befindet sich in der Vorschauversion für USA Ost (Nord-Virginia), USA West (Oregon) und Asien-Pazifik (Tokio) für CloudWatch und kann sich ändern.

CloudWatch unterstützt eine Abfragefunktion in natürlicher Sprache, mit der Sie Abfragen für CloudWatch Metrics Insights und CloudWatch Logs Insights generieren und aktualisieren können.

Mit dieser Funktion können Sie Fragen zu den CloudWatch Daten, nach denen Sie suchen, stellen oder sie in einfachem Englisch beschreiben. Die Funktion in natürlicher Sprache generiert eine Abfrage auf der Grundlage einer von Ihnen eingegebenen Eingabeaufforderung und bietet eine line-by-line Erläuterung der Funktionsweise der Abfrage. Sie können Ihre Abfrage auch aktualisieren, um Ihre Daten weiter zu untersuchen.

Abhängig von Ihrer Umgebung können Sie Eingabeaufforderungen wie „Welche Instance in Amazon Elastic Compute Cloud hat den höchsten Netzwerkausgang?“ und „Zeigt mir die zehn Amazon DynamoDB-Tabellen mit den meisten verbrauchten Lesevorgängen.“ eingeben.

Um eine CloudWatch Metrics Insights-Abfrage mit dieser Funktion zu generieren, öffnen Sie den CloudWatch Metrics Insights-Abfrage-Editor in der Builder - oder Editor-Ansicht und wählen Sie Abfrage generieren aus.

> **Important**
>
> Um die Abfragefunktion in natürlicher Sprache zu verwenden, müssen Sie die ReadOnlyAccessRichtlinie CloudWatchFullAccessCloudWatchReadOnlyAccess,, CloudWatchFullAccessV2 AdministratorAccess, oder verwenden.
> Sie können die `cloudwatch:GenerateQuery`-Aktion auch in eine neue oder bestehende, vom Kunden verwaltete oder integrierte Richtlinie aufnehmen.

Beispielabfragen

In den Beispielen in diesem Abschnitt wird beschrieben, wie Abfragen mithilfe der natürlichen Sprachfunktion generiert und aktualisiert werden.

> **Note**
>
> Weitere Informationen zum Abfrageeditor und zur Syntax von CloudWatch Metrics Insights finden Sie unter CloudWatch Metrics Insights-Abfragekomponenten und Syntax.

Beispiel: Eine Abfrage in natürlicher Sprache generieren

Um eine Abfrage in natürlicher Sprache zu generieren, geben Sie eine Aufforderung ein und wählen Sie Neue Abfrage generieren. Dieses Beispiel zeigt eine Abfrage, die eine einfache Suche durchführt.

Telefonansage

Im Folgenden finden Sie ein Beispiel für eine Eingabeaufforderung, die die Fähigkeit anweist, nach den 10 DynamoDB-Tabellen zu suchen, die die meiste Lesekapazität verbrauchen.

```
Show top 10 DynamoDB Tables by consumed reads
```

Abfrage

Im Folgenden finden Sie ein Beispiel für eine Abfrage, die die Funktion natürlicher Sprache anhand der Eingabeaufforderung generiert. Beachten Sie, wie die Aufforderung in einem Kommentar vor der Abfrage erscheint. Nach der Abfrage können Sie eine Erklärung lesen, in der beschrieben wird, wie die Abfrage funktioniert.

```
# Show top 10 DynamoDB Tables by consumed reads
SELECT SUM("ConsumedReadCapacityUnits")
FROM "AWS/DynamoDB"
GROUP BY TableName
ORDER BY SUM() DESC
LIMIT 10
# This query selects the sum of consumed read capacity units for each DynamoDB table,
 groups the results by table name, orders the results from highest to lowest read
 capacity consumption, and limits the results to the top 10 tables.
```

> **Note**
>
> Verwenden Sie das Zahnradsymbol in Ihrem Editor, um das Erscheinungsbild Ihrer Aufforderung und die Erläuterung der Funktionsweise der Abfrage zu deaktivieren.

Beispiel: Eine Abfrage in natürlicher Sprache aktualisieren

Sie können eine Abfrage aktualisieren, indem Sie die erste Eingabeaufforderung bearbeiten und dann Abfrage aktualisieren wählen.

Aktualisierte Eingabeaufforderung

Das folgende Beispiel zeigt eine aktualisierte Version der vorherigen Eingabeaufforderung. Anstatt einer Eingabeaufforderung, die nach den Top 10 DynamoDB-Tabellen sucht, die die meiste Lesekapazität verbrauchen, leitet diese Aufforderung nun die Funktion an, die Ergebnisse nach der Anzahl der zurückgegebenen Byte zu sortieren.

```
Sort by bytes returned instead
```

Aktualisierte Abfrage

Im Folgenden finden Sie ein Beispiel für die aktualisierte Abfrage. Beachten Sie, wie die Eingabeaufforderung in einem Kommentar vor der Abfrage erscheint. Nach der Abfrage können Sie eine Erklärung lesen, in der beschrieben wird, wie die ursprüngliche Abfrage aktualisiert wurde.

```
# Sort by bytes returned instead
SELECT SUM("ReturnedBytes")
FROM "AWS/DynamoDB"
GROUP BY TableName
ORDER BY SUM() DESC
LIMIT 10
# This query modifies the original query to select the sum of returned bytes instead
 of consumed read capacity units, and orders the results from highest to lowest sum of
 returned bytes, limiting the results to the top 10 tables.
```

Abmeldung von der Verwendung Ihrer Daten zur Serviceverbesserung

Die Eingabeaufforderungs-Daten in natürlicher Sprache, die Sie bereitstellen, um das KI-Modell zu trainieren und relevante Abfragen zu generieren, werden ausschließlich zur Bereitstellung und Wartung Ihres Services verwendet. Diese Daten können verwendet werden, um die Qualität von CloudWatch Metrics Insights zu verbessern. Ihr Vertrauen, Ihre Privatsphäre sowie die Sicherheit Ihrer Inhalte sind unsere obersten Prioritäten. Weitere Informationen finden Sie unter AWS -Service-Bedingungen und AWS verantwortliche KI-Richtlinie.

Sie können die Verwendung Ihrer Inhalte zur Entwicklung oder Verbesserung der Qualität von Abfragen in natürlicher Sprache deaktivieren, indem Sie eine Opt-Out-Richtlinie für KI-Services erstellen. Um die Datenerfassung für alle CloudWatch KI-Funktionen, einschließlich der Funktion zur Abfragegenerierung, abzulehnen, müssen Sie eine Opt-Out-Richtlinie für erstellen CloudWatch. Weitere Informationen finden Sie unter Opt-Out-Richtlinien für KI-Services im Benutzerhandbuch für AWS Organizations .

SQL-Inferenz

CloudWatch Metrics Insights verwendet mehrere Mechanismen, um auf die Absicht einer bestimmten SQL-Abfrage zu schließen.

Themen

- Zeit-Bucketing
- Projektion von Feldern
- SORTIEREN NACH globaler Aggregation

Zeit-Bucketing

Zeitreihendatenpunkte, die sich aus einer Abfrage ergeben, werden basierend auf dem angeforderten Zeitraum in Zeitbuckets zusammengeführt. Um Werte in Standard-SQL zu aggregieren, muss eine explizite GROUP BY-Klausel definiert werden, um alle Beobachtungen eines bestimmten Zeitraums zusammen zu sammeln. Da dies die Standardmethode für die Abfrage von Zeitreihendaten ist, leitet CloudWatch Metrics Insights eine Zeitspanne ab, ohne dass eine explizite GROUP BY-Klausel ausgedrückt werden muss.

Wenn beispielsweise eine Abfrage mit einem Zeitraum von einer Minute ausgeführt wird, werden alle Beobachtungen, die zu dieser Minute bis zur nächsten (ausgeschlossen) gehören, bis zur Startzeit des Zeitbuckets zusammengeführt. Dies macht Metrics-Insights-SQL-Anweisungen prägnanter und weniger ausführlich.

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
```

Die vorherige Abfrage gibt eine einzelne Zeitreihe (Zeitstempel-Wert-Paare) zurück, die die durchschnittliche CPU-Auslastung aller Amazon-EC2-Instances darstellt. Unter der Annahme, dass der angeforderte Zeitraum eine Minute beträgt, stellt jeder zurückgegebene Datenpunkt den Durchschnitt aller Beobachtungen dar, die innerhalb eines bestimmten Intervalls von einer Minute gemessen werden (Startzeit inklusive, Endzeit exklusiv). Der Zeitstempel, der sich auf den spezifischen Datenpunkt bezieht, ist die Startzeit des Buckets

Projektion von Feldern

Metrics-Insights-Abfragen geben immer die Zeitstempelprojektion zurück. Sie müssen keine Zeitstempelspalte in der SELECT-Klausel angeben, um den Zeitstempel jedes entsprechenden Datenpunktwerts abzurufen. Weitere Informationen dazu, wie der Zeitstempel berechnet wird, finden Sie unter Zeit-Bucketing.

Bei Verwendung von GROUP BY wird jeder Gruppenname ebenfalls abgeleitet und im Ergebnis projiziert, sodass Sie die zurückgegebenen Zeitreihen gruppieren können.

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
GROUP BY InstanceId
```

Die vorherige Abfrage gibt eine Zeitreihe für jede Amazon-EC2-Instance zurück. Jede Zeitreihe wird nach dem Wert der Instance-ID beschriftet.

SORTIEREN NACH globaler Aggregation

Wenn Sie ORDER BY verwenden, leitet FUNCTION() ab, nach welcher Aggregatfunktion Sie sortieren möchten (die Datenpunktwerte der abgefragten Metriken). Der Aggregatvorgang wird über alle übereinstimmenden Datenpunkte jeder einzelnen Zeitreihe im abgefragten Zeitfenster ausgeführt.

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
GROUP BY InstanceId
ORDER BY MAX()
LIMIT 10
```

Die vorherige Abfrage gibt die CPU-Auslastung für jede Amazon-EC2-Instance zurück und beschränkt die Ergebnismenge auf 10 Einträge. Die Ergebnisse werden basierend auf dem Maximalwert der einzelnen Zeitreihe innerhalb des angeforderten Zeitfensters sortiert. Die ORDER BY-Klausel wird vor LIMIT angewendet, damit die Sortierung mit mehr als 10 Zeitreihen berechnet wird.

Beispielabfragen für Metriken Insights

Dieser Abschnitt enthält Beispiele für nützliche CloudWatch Metrics Insights-Abfragen, die Sie kopieren und direkt verwenden oder im Abfrage-Editor kopieren und ändern können. Einige dieser Beispiele sind bereits in der Konsole verfügbar und Sie können auf sie zugreifen, indem Sie in der Metrikanzeige Add query (Abfrage hinzufügen) auswählen.

Beispiele für Application Load Balancer

Gesamtzahl der Anfragen für alle Load Balancer

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer)
```

Top 10 der aktivsten Load Balancer

```
SELECT MAX(ActiveConnectionCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer)
GROUP BY LoadBalancer
ORDER BY SUM() DESC
LIMIT 10
```

AWS Beispiele für die API-Nutzung

Die 20 wichtigsten AWS APIs nach der Anzahl der Aufrufe in Ihrem Konto

```
SELECT COUNT(CallCount)
FROM SCHEMA("AWS/Usage", Class, Resource, Service, Type)
WHERE Type = 'API'
GROUP BY Service, Resource
ORDER BY COUNT() DESC
LIMIT 20
```

CloudWatch APIs, sortiert nach Aufrufen

```
SELECT COUNT(CallCount)
FROM SCHEMA("AWS/Usage", Class, Resource, Service, Type)
WHERE Type = 'API' AND Service = 'CloudWatch'
GROUP BY Resource
ORDER BY COUNT() DESC
```

DynamoDB-Beispiele

Top 10 Tabellen nach verbrauchten Lesevorgängen

```
SELECT SUM(ProvisionedWriteCapacityUnits)
FROM SCHEMA("AWS/DynamoDB", TableName)
GROUP BY TableName
ORDER BY MAX() DESC LIMIT 10
```

Top 10 Tabellen nach zurückgegebenen Bytes

```
SELECT SUM(ReturnedBytes)
FROM SCHEMA("AWS/DynamoDB", TableName)
GROUP BY TableName
ORDER BY MAX() DESC LIMIT 10
```

Top 10 Tabellen nach Benutzerfehlern

```
SELECT SUM(UserErrors)
FROM SCHEMA("AWS/DynamoDB", TableName)
GROUP BY TableName
ORDER BY MAX() DESC LIMIT 10
```

Beispiele für Amazon Elastic Block Store

Top 10 Amazon-EBS-Volumes nach geschriebenen Bytes

```
SELECT SUM(VolumeWriteBytes)
FROM SCHEMA("AWS/EBS", VolumeId)
GROUP BY VolumeId
ORDER BY SUM() DESC
LIMIT 10
```

Durchschnittliche Schreibzeit des Amazon-EBS-Volumes

```
SELECT AVG(VolumeTotalWriteTime)
FROM SCHEMA("AWS/EBS", VolumeId)
```

Beispiele für Amazon EC2

CPU-Auslastung von EC2-Instances sortiert nach den höchsten Auslastungen

```
SELECT AVG(CPUUtilization)
  FROM SCHEMA("AWS/EC2", InstanceId)
  GROUP BY InstanceId
  ORDER BY AVG() DESC
```

Durchschnittliche CPU-Auslastung für die gesamte Flotte

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
```

Top 10 Instances nach höchster CPU-Auslastung

```
SELECT MAX(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
```

```
GROUP BY InstanceId
ORDER BY MAX() DESC
LIMIT 10
```

In diesem Fall erfasst der CloudWatch Agent eine **`CPUUtilization`** Metrik pro Anwendung. Diese Abfrage filtert den Durchschnitt dieser Metrik für einen bestimmten Anwendungsnamen.

```
SELECT AVG(CPUUtilization)
FROM "AWS/CWAgent"
WHERE ApplicationName = 'eCommerce'
```

Beispiele für Amazon Elastic Container Service

Durchschnittliche CPU-Auslastung in allen ECS-Clustern

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/ECS", ClusterName)
```

Top 10 Cluster nach Speicherauslastung

```
SELECT AVG(MemoryUtilization)
FROM SCHEMA("AWS/ECS", ClusterName)
GROUP BY ClusterName
ORDER BY AVG() DESC
LIMIT 10
```

Top 10 Services nach CPU-Auslastung

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/ECS", ClusterName, ServiceName)
GROUP BY ClusterName, ServiceName
ORDER BY AVG() DESC
LIMIT 10
```

Top 10 Services durch Ausführen von Aufgaben (Container Insights)

```
SELECT AVG(RunningTaskCount)
FROM SCHEMA("ECS/ContainerInsights", ClusterName, ServiceName)
GROUP BY ClusterName, ServiceName
ORDER BY AVG() DESC
```

```
LIMIT 10
```

Beispiele für Amazon Elastic Kubernetes Service Container Insights

Durchschnittliche CPU-Auslastung in allen EKS-Clustern

```
SELECT AVG(pod_cpu_utilization)
FROM SCHEMA("ContainerInsights", ClusterName)
```

Top 10 Cluster nach Knoten-CPU-Auslastung

```
SELECT AVG(node_cpu_utilization)
FROM SCHEMA("ContainerInsights", ClusterName)
GROUP BY ClusterName
ORDER BY AVG() DESC LIMIT 10
```

Top 10 Cluster nach Pod-Speicherauslastung

```
SELECT AVG(pop_memory_utilization)
FROM SCHEMA("ContainerInsights", ClusterName)
GROUP BY ClusterName
ORDER BY AVG() DESC LIMIT 10
```

Top 10 Knoten nach CPU-Auslastung

```
SELECT AVG(node_cpu_utilization)
FROM SCHEMA("ContainerInsights", ClusterName, NodeName)
GROUP BY ClusterName, NodeName
ORDER BY AVG() DESC LIMIT 10
```

Top 10 Pods nach Speicherauslastung

```
SELECT AVG(pod_memory_utilization)
FROM SCHEMA("ContainerInsights", ClusterName, PodName)
GROUP BY ClusterName, PodName
ORDER BY AVG() DESC LIMIT 10
```

EventBridge Beispiele

Top 10 Regeln nach Aufrufen

```
SELECT SUM(Invocations)
FROM SCHEMA("AWS/Events", RuleName)
GROUP BY RuleName
ORDER BY MAX() DESC LIMIT 10
```

Top 10 Regeln nach fehlgeschlagenen Aufrufen

```
SELECT SUM(FailedInvocations)
FROM SCHEMA("AWS/Events", RuleName)
GROUP BY RuleName
ORDER BY MAX() DESC LIMIT 10
```

Top 10 Regeln nach übereinstimmenden Regeln

```
SELECT SUM(MatchedEvents)
FROM SCHEMA("AWS/Events", RuleName)
GROUP BY RuleName
ORDER BY MAX() DESC LIMIT 10
```

Beispiele für Kinesis

Top 10 Streams nach geschriebenen Bytes

```
SELECT SUM("PutRecords.Bytes")
FROM SCHEMA("AWS/Kinesis", StreamName)
GROUP BY StreamName
ORDER BY SUM() DESC LIMIT 10
```

Top 10 Streams nach frühesten Items im Stream

```
SELECT MAX("GetRecords.IteratorAgeMilliseconds")
FROM SCHEMA("AWS/Kinesis", StreamName)
GROUP BY StreamName
ORDER BY MAX() DESC LIMIT 10
```

Beispiele für Lambda

Lambda-Funktionen geordnet nach Anzahl der Aufrufe

```
SELECT SUM(Invocations)
```

```
FROM SCHEMA("AWS/Lambda", FunctionName)
GROUP BY FunctionName
ORDER BY SUM() DESC
```

Top 10 Lambda-Funktionen bei längster Laufzeit

```
SELECT AVG(Duration)
FROM SCHEMA("AWS/Lambda", FunctionName)
GROUP BY FunctionName
ORDER BY MAX() DESC
LIMIT 10
```

Top 10 Lambda-Funktionen nach Fehleranzahl

```
SELECT SUM(Errors)
FROM SCHEMA("AWS/Lambda", FunctionName)
GROUP BY FunctionName
ORDER BY SUM() DESC
LIMIT 10
```

CloudWatch Logs, Beispiele

Top 10 Protokollgruppen nach eingehenden Ereignissen

```
SELECT SUM(IncomingLogEvents)
FROM SCHEMA("AWS/Logs", LogGroupName)
GROUP BY LogGroupName
ORDER BY SUM() DESC LIMIT 10
```

Top 10 Protokollgruppen nach geschriebenen Bytes

```
SELECT SUM(IncomingBytes)
FROM SCHEMA("AWS/Logs", LogGroupName)
GROUP BY LogGroupName
ORDER BY SUM() DESC LIMIT 10
```

Beispiele für Amazon RDS

Top 10 Amazon-RDS-Instances nach höchster CPU-Auslastung

```
SELECT MAX(CPUUtilization)
```

```
FROM SCHEMA("AWS/RDS", DBInstanceIdentifier)
GROUP BY DBInstanceIdentifier
ORDER BY MAX() DESC
LIMIT 10
```

Top 10 Amazon-RDS-Cluster nach Schreibvorgängen

```
SELECT SUM(WriteIOPS)
FROM SCHEMA("AWS/RDS", DBClusterIdentifier)
GROUP BY DBClusterIdentifier
ORDER BY MAX() DESC
LIMIT 10
```

Beispiele für Amazon Simple Storage Service

Durchschnittliche Latenz nach Bucket

```
SELECT AVG(TotalRequestLatency)
FROM SCHEMA("AWS/S3", BucketName, FilterId)
WHERE FilterId = 'EntireBucket'
GROUP BY BucketName
ORDER BY AVG() DESC
```

Top 10 Buckets nach heruntergeladenen Bytes

```
SELECT SUM(BytesDownloaded)
FROM SCHEMA("AWS/S3", BucketName, FilterId)
WHERE FilterId = 'EntireBucket'
GROUP BY BucketName
ORDER BY SUM() DESC
LIMIT 10
```

Beispiele für Amazon Simple Notification Service

Gesamtzahl der von SNS-Themen veröffentlichten Nachrichten

```
SELECT SUM(NumberOfMessagesPublished)
FROM SCHEMA("AWS/SNS", TopicName)
```

Top 10 Themen nach veröffentlichten Nachrichten

```
SELECT SUM(NumberOfMessagesPublished)
FROM SCHEMA("AWS/SNS", TopicName)
GROUP BY TopicName
ORDER BY SUM() DESC
LIMIT 10
```

Top 10 Themen nach Fehlern bei der Nachrichtenübermittlung

```
SELECT SUM(NumberOfNotificationsFailed)
FROM SCHEMA("AWS/SNS", TopicName)
GROUP BY TopicName
ORDER BY SUM() DESC
LIMIT 10
```

Beispiele für Amazon SQS

Die 10 häufigsten Warteschlangen nach Anzahl sichtbarer Nachrichten

```
SELECT AVG(ApproximateNumberOfMessagesVisible)
FROM SCHEMA("AWS/SQS", QueueName)
GROUP BY QueueName
ORDER BY AVG() DESC
LIMIT 10
```

Die 10 aktivsten Warteschlangen

```
SELECT SUM(NumberOfMessagesSent)
FROM SCHEMA("AWS/SQS", QueueName)
GROUP BY QueueName
ORDER BY SUM() DESC
LIMIT 10
```

Top 10 Warteschlangen nach Alter der ersten Nachricht

```
SELECT AVG(ApproximateAgeOfOldestMessage)
FROM SCHEMA("AWS/SQS", QueueName)
GROUP BY QueueName
ORDER BY AVG() DESC
LIMIT 10
```

Limits für Metric Insights

CloudWatch Metrics Insights hat derzeit die folgenden Beschränkungen:

- Derzeit können Sie nur die letzten drei Stunden an Daten abfragen.
- Eine einzelne Abfrage kann nicht mehr als 10 000 Metriken verarbeiten. Dies bedeutet, wenn die Klauseln SELECT, FROM und WHERE mehr als 10 000 Metriken entsprechen, die Abfrage nur die ersten 10 000 der gefundenen Metriken verarbeitet.
- Eine einzelne Abfrage kann nicht mehr als 500 Zeitreihen zurückgeben. Dies bedeutet, wenn die Abfrage mehr als 500 Metriken zurückgibt, nicht alle Metriken in den Abfrageergebnissen zurückgegeben werden. Wenn Sie eine ORDER BY-Klausel verwenden, werden alle zu verarbeitenden Metriken sortiert, und die 500, die nach der ORDER BY-Klausel den höchsten oder niedrigsten Wert haben, werden zurückgegeben.

  Wenn Sie keine ORDER BY-Klausel einschließen, können Sie nicht steuern, welche 500 übereinstimmenden Metriken zurückgegeben werden.
- Sie können bis zu 200 Metrics Insights-Alarme pro Region einrichten.
- Metrics Inights unterstützt keine hochauflösenden Daten, wenn es sich um metrische Daten handelt, die mit einer Granularität von weniger als einer Minute gemeldet werden. Wenn Sie hochauflösende Daten anfordern, schlägt die Anforderung nicht fehl, sondern die Ausgabe wird mit einer Granularität von einer Minute aggregiert.
- Jeder GetMetricDataVorgang kann nur eine Abfrage haben, aber Sie können mehrere Widgets in einem Dashboard haben, die jeweils eine Abfrage enthalten.

Glossar zu Metric Insights

Bezeichnung

In Metrics Insights ist ein Bezeichner ein Schlüssel-Wert-Paar, das verwendet wird, um eine Abfrage zu veranlassen, einen bestimmten Datensatz zurückzugeben oder um Kriterien zu definieren, durch die Abfrageergebnisse in separate Zeitreihen unterteilt werden sollen. Ein Bezeichnungsschlüssel ähnelt einem Spaltennamen in SQL. Derzeit müssen Beschriftungen CloudWatch metrische Dimensionen haben.

Beobachtung

Eine Beobachtung ist ein Wert, der zu einem bestimmten Zeitpunkt für eine bestimmte Metrik aufgezeichnet wurde.

Problembehebung bei Metrics Insights

Die Ergebnisse beinhalten „Andere“, aber ich habe das nicht als Dimension

Dies bedeutet, dass die Abfrage eine GROUP BY-Klausel enthält, die einen Bezeichnungsschlüssel angibt, der in einigen der Metriken, die von der Abfrage zurückgegeben werden, nicht verwendet wird. In diesem Fall wird eine Nullgruppe mit dem Namen `Other` zurückgegeben. Die Metriken, die diesen Bezeichnungsschlüssel nicht enthalten, sind wahrscheinlich aggregierte Metriken, die Werte zurückgeben, die über alle Werte dieses Bezeichnungsschlüssels aggregiert sind.

Angenommen, wir haben die folgende Abfrage:

```
SELECT AVG(Faults)
FROM MyCustomNamespace
GROUP BY Operation, ServiceName
```

Wenn beispielsweise einige der zurückgegebenen Metriken `ServiceName` nicht als Dimension enthalten, werden diese Metriken so angezeigt, als hätte sie `Other` als Wert für `ServiceName`.

Um zu verhindern, dass „Andere“ in Ihren Ergebnissen angezeigt wird, verwenden Sie SCHEMA in Ihrer FROM-Klausel, wie im folgenden Beispiel dargestellt:

```
SELECT AVG(Faults)
FROM SCHEMA(MyCustomNamespace, Operation)
GROUP BY Operation, ServiceName
```

Dies beschränkt die zurückgegebenen Ergebnisse nur auf die Metriken, die über die Dimensionen `Operation` und `ServiceName` verfügen.

Der älteste Zeitstempel in meinem Diagramm hat einen niedrigeren Metrikwert als die anderen

CloudWatch Metrics Insights unterstützt derzeit nur Daten der letzten drei Stunden. Wenn Sie ein Diagramm mit einer Periode von mehr als einer Minute erstellen, kann es Fälle geben, in denen der älteste Datenpunkt vom erwarteten Wert abweicht. Das liegt daran, dass die Metrics Insights-Abfragen nur die Daten der letzten drei Stunden zurückgeben. In diesem Fall gibt der älteste Datenpunkt in der Abfrage nur die Beobachtungen zurück, die innerhalb der letzten drei Stunden gemessen wurden, anstatt alle Beobachtungen innerhalb des Zeitraums dieses Datenpunkts zurückzugeben.

Verwenden Sie den Metrik-Explorer, um Ressourcen anhand ihrer Tags und Eigenschaften zu überwachen

Metrik-Explorer ist ein tag-basiertes Tool, mit dem Sie Ihre Metriken nach Tags und Ressourceneigenschaften filtern, aggregieren und visualisieren können, um die Beobachtbarkeit Ihrer Services zu verbessern. Dadurch erhalten Sie eine flexible und dynamische Problembehandlung, sodass Sie mehrere Diagramme gleichzeitig erstellen und diese Diagramme zum Erstellen Ihrer Anwendungsintegritäts-Dashboards verwenden können.

Visualisierungen im Metrik-Explorer sind dynamisch. Wenn also eine passende Ressource erstellt wird, nachdem Sie ein Metrik-Explorer-Widget erstellt und zu einem CloudWatch Dashboard hinzugefügt haben, wird die neue Ressource automatisch im Explorer-Widget angezeigt.

Wenn beispielsweise alle Ihre EC2-Produktionsinstances über das **`production`**-Tag verfügen, können Sie den Metrik-Explorer verwenden, um Metriken von all diesen Instances zu filtern und zu aggregieren, um deren Zustand und Leistung zu verstehen. Wenn später eine neue Instance mit einem übereinstimmenden Tag erstellt wird, wird sie automatisch dem Metrik-Explorer-Widget hinzugefügt.

> **Note**
>
> Der Metrics Explorer bietet ein point-in-time Erlebnis. Ressourcen, die beendet wurden oder mit der von Ihnen angegebenen Eigenschaft oder dem Tag nicht mehr existieren, werden in der Visualisierung nicht angezeigt. Sie können die Metriken für diese Ressourcen jedoch weiterhin in den CloudWatch Metrikansichten finden.

Mit dem Metrik-Explorer können Sie festlegen, wie Metriken aus den Ressourcen aggregiert werden, die den Kriterien entsprechen, und ob sie alle in einem einzigen Diagramm oder in verschiedenen Diagrammen innerhalb eines Metrik-Explorer-Widgets angezeigt werden sollen.

Der Metrik-Explorer enthält Vorlagen, die Sie verwenden können, um nützliche Visualisierungsdiagramme mit einem Klick anzuzeigen, und Sie können diese Vorlagen auch erweitern, um vollständig angepasste Metrik-Explorer-Widgets zu erstellen.

Der Metrics Explorer unterstützt Metriken, die vom Agenten ausgegeben werden, AWS und EC2-Metriken, die vom CloudWatch Agenten veröffentlicht werden, einschließlich Speicher-, Festplatten- und CPU-Metriken. Um den Metrics Explorer zu verwenden, um die vom CloudWatch

Agenten veröffentlichten Metriken zu sehen, müssen Sie möglicherweise Ihre CloudWatch Agenten-Konfigurationsdatei aktualisieren. Weitere Informationen finden Sie unter CloudWatch Agentenkonfiguration für den Metrik-Explorer.

Gehen Sie folgendermaßen vor, um eine Visualisierung mit Metrik-Explorer zu erstellen und sie optional einem Dashboard hinzuzufügen.

So erstellen Sie eine Visualisierung mit Metrik-Explorer

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Führen Sie eine der folgenden Aktionen aus:
   - Um eine Vorlage zu verwenden, wählen Sie sie in dem Feld aus, das derzeit den leeren Explorer anzeigt.

     Abhängig von der Vorlage zeigt der Explorer möglicherweise sofort Diagramme von Metriken an. Wenn dies nicht der Fall ist, wählen Sie im Feld Von ein oder mehrere Tags oder Eigenschaften aus, und dann sollten Daten angezeigt werden. Wenn dies nicht der Fall ist, verwenden Sie die Optionen oben auf der Seite, um einen längeren Zeitraum in den Diagrammen anzuzeigen.
   - Um eine benutzerdefinierte Visualisierung zu erstellen, wählen Sie unter Metriken eine einzelne Metrik oder alle verfügbaren Metriken aus einem Service aus.

     Nachdem Sie eine Metrik ausgewählt haben, können Sie diesen Schritt optional wiederholen, um weitere Metriken hinzuzufügen.
4. CloudWatch Zeigt für jede ausgewählte Metrik unmittelbar nach dem Metriknamen die Statistik an, die verwendet wird. Wählen Sie unter Wählen Sie den Statistiknamen und anschließend die gewünschte Statistik aus.
5. Wählen Sie unter Von ein Tag oder eine Ressourceneigenschaft aus, um Ihre Ergebnisse zu filtern.

   Danach können Sie diesen Schritt optional wiederholen, um weitere Tags oder Ressourceneigenschaften auszuwählen.

   Wenn Sie mehrere Werte derselben Eigenschaft auswählen, z. B. zwei EC2-Instance-Typen, zeigt der Explorer alle Ressourcen an, die mit den ausgewählten Eigenschaften übereinstimmen. Es wird als ODER-Operation behandelt.

Wenn Sie andere Eigenschaften oder Tags auswählen, z. B. das **`Production`**-Tag und den M5-Instance-Typ, werden nur die Ressourcen angezeigt, die allen diesen Auswahlen entsprechen. Dies wird als UND-Operation behandelt.

6. (Optional) Wählen Sie für Aggregieren nach eine Statistik aus, die zum Aggregieren der Metriken verwendet werden soll. Wählen Sie dann neben für aus, wie die Metrik aus der Liste aggregiert werden soll. Sie können alle Ressourcen zusammenfassen, die derzeit angezeigt werden, oder durch ein einzelnes Tag oder eine Ressourceneigenschaft aggregieren.

   Je nachdem, wie Sie sich für die Aggregierung entscheiden, kann das Ergebnis eine einzelne Zeitreihe oder mehrere Zeitreihen sein.

7. Unter Aufteilen nach können Sie ein einzelnes Diagramm mit mehreren Zeitreihen in mehrere Diagramme aufteilen. Die Aufteilung kann nach verschiedenen Kriterien erfolgen, die Sie unter Aufteilen nach auswählen.

8. Unter Diagrammoptionen können Sie das Diagramm verfeinern, indem Sie den Zeitraum, den Diagrammtyp, die Legendenplatzierung und das Layout ändern.

9. Um diese Visualisierung als Widget zu einem CloudWatch Dashboard hinzuzufügen, wählen Sie Zum Dashboard hinzufügen.

CloudWatch Agentenkonfiguration für den Metrik-Explorer

Damit der Metrics Explorer die vom CloudWatch Agenten veröffentlichten EC2-Metriken erkennen kann, stellen Sie sicher, dass die CloudWatch Agenten-Konfigurationsdatei die folgenden Werte enthält:

- Stellen Sie im `metrics`-Abschnitt sicher, dass der `aggregation_dimensions`-Parameter `["InstanceId"]` enthält. Er kann auch andere Dimensionen enthalten.
- Stellen Sie im `metrics`-Abschnitt sicher, dass der `append_dimensions`-Parameter eine `{"InstanceId":"${aws:InstanceId}"}`-Zeile enthält. Er kann auch andere Zeilen enthalten.
- Überprüfen Sie im Abschnitt `metrics` innerhalb des Abschnitts `metrics_collected` die Abschnitte für jeden Ressourcentyp, den der Metrik-Explorer ermitteln soll, z. B. die Abschnitte `cpu`, `disk` und `memory`. Stellen Sie sicher, dass jeder dieser Abschnitte einen `"resources": [ "*"] line.` enthält.
- Stellen Sie im `cpu`-Abschnitt des `metrics_collected`-Abschnitts sicher, dass eine `"totalcpu": true`-Zeile vorhanden ist.

- Sie müssen den `CWAgent` Standard-Namespace für die vom CloudWatch Agenten gesammelten Metriken anstelle eines benutzerdefinierten Namespaces verwenden.

Die Einstellungen in der vorherigen Liste veranlassen den CloudWatch Agenten, aggregierte Messwerte für Festplatten, CPUs und andere Ressourcen zu veröffentlichen, die im Metrik-Explorer für alle Instanzen, die ihn verwenden, dargestellt werden können.

Durch diese Einstellungen werden die Metriken, die Sie zuvor für die Veröffentlichung mit mehreren Dimensionen eingerichtet hatten, erneut veröffentlicht, wodurch sich Ihre Metrikkosten erhöhen.

Weitere Hinweise zur Bearbeitung der CloudWatch Agenten-Konfigurationsdatei finden Sie unter. Erstellen oder bearbeiten Sie die CloudWatch Agenten-Konfigurationsdatei manuell

Metrik-Streams verwenden

Sie können Metrik-Streams verwenden, um CloudWatch Metriken kontinuierlich und mit near-real-time geringer Latenz an ein Ziel Ihrer Wahl zu streamen. Zu den unterstützten AWS Zielen gehören Ziele wie Amazon Simple Storage Service und mehrere Ziele von Drittanbietern.

Es gibt drei Hauptnutzungsszenarien für CloudWatch Metrik-Streams:

- Benutzerdefiniertes Setup mit Firehose — Erstellen Sie einen Metrik-Stream und leiten Sie ihn an einen Amazon Data Firehose-Lieferstream weiter, der Ihre CloudWatch Metriken dorthin liefert, wo Sie sie haben möchten. Sie können sie an einen Data Lake wie Amazon S3 oder an jedes Ziel oder Endgerät streamen, das von Firehose unterstützt wird, einschließlich Drittanbietern. Die Formate JSON, OpenTelemetry 1.0.0 und OpenTelemetry 0.7.0 werden nativ unterstützt, oder Sie können Transformationen in Ihrem Firehose-Lieferstream konfigurieren, um die Daten in ein anderes Format wie Parquet zu konvertieren. Mit einem Metrik-Stream können Sie die Überwachungsdaten kontinuierlich aktualisieren oder diese CloudWatch Metrikdaten mit Abrechnungs- und Leistungsdaten kombinieren, um umfangreiche Datensätze zu erstellen. Sie können dann Tools wie Amazon Athena verwenden, um eine Erkenntnis in die Kostenoptimierung, Ressourcenleistung und Ressourcenauslastung zu erhalten.
- S3 Quick Setup – Streamen Sie mit einem Quick-Setup-Verfahren zu Amazon Simple Storage Service. CloudWatch Erstellt standardmäßig die für den Stream benötigten Ressourcen. Die Formate JSON, OpenTelemetry 1.0.0 und OpenTelemetry 0.7.0 werden unterstützt.
- Schnelles AWS Partner-Setup — CloudWatch bietet einigen Drittanbietern eine schnelle Einrichtung. Sie können Drittanbieter verwenden, um Ihre Anwendungen mithilfe der gestreamten CloudWatch Daten zu überwachen, Fehler zu beheben und zu analysieren. Wenn Sie den

schnellen Partner-Setup-Workflow verwenden, müssen Sie nur eine Ziel-URL und einen API-Schlüssel für Ihr Ziel angeben und kümmern CloudWatch sich um den Rest der Einrichtung. Quick Setup für Partner ist für die folgenden Drittanbieter verfügbar:

- Datadog
- Dynatrace
- New Relic
- Splunk Observability Cloud
- SumoLogic

Sie können alle Ihre CloudWatch Metriken streamen oder Filter verwenden, um nur bestimmte Metriken zu streamen. Jeder Metrik-Stream kann bis zu 1 000 Filter enthalten, die entweder Metrik-Namespaces oder bestimmte Metriken einschließen oder ausschließen. Ein einzelner Metrik-Stream kann nur Ein- oder Ausschlussfilter haben, aber nicht beide.

Wenn nach dem Erstellen eines Metrik-Streams neue Metriken erstellt werden, die mit den vorhandenen Filtern übereinstimmen, werden die neuen Metriken automatisch in den Stream aufgenommen.

Es gibt keine Begrenzung für die Anzahl der Metrik-Streams pro Konto oder pro Region und keine Begrenzung für die Anzahl der Metrik-Aktualisierungen, die gestreamt werden.

Jeder Stream kann entweder das JSON-Format, das OpenTelemetry 1.0.0- oder das OpenTelemetry 0.7.0-Format verwenden. Sie können das Ausgabeformat eines Metrik-Streams jederzeit bearbeiten, z. B. für ein Upgrade von OpenTelemetry 0.7.0 auf 1.0.0. OpenTelemetry Weitere Informationen zu Ausgabeformaten finden Sie unter Ausgabeformate für Metrik-Streams.

Für Metrik-Streams in Überwachungskonten können Sie wählen, ob Metriken aus den mit diesem Überwachungskonto verknüpften Quellkonten einbezogen werden sollen. Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.

Metrikstreams beinhalten immer die Statistiken `Minimum`, `Maximum`, `SampleCount` und `Sum`. Gegen Aufpreis können Sie auch zusätzliche Statistiken angeben. Weitere Informationen finden Sie unter Statistiken, die gestreamt werden können.

Die Preise für Metrik-Streams basieren auf der Anzahl der Metrik-Aktualisierungen. Firehose berechnet Ihnen auch Gebühren für den Delivery Stream, der für den Metric Stream verwendet wird. Weitere Informationen finden Sie unter CloudWatch Amazon-Preise.

Themen

Einen Metrik-Stream einrichten

Verwenden Sie die Schritte in den folgenden Abschnitten, um einen CloudWatch Metrik-Stream einzurichten.

Nachdem ein Metrik-Stream erstellt wurde, hängt die Zeit, die benötigt wird, bis Metrikdaten am Ziel angezeigt werden, von den konfigurierten Puffereinstellungen im Firehose-Lieferstream ab. Die Pufferung wird in der maximalen Nutzlastgröße oder der maximalen Wartezeit ausgedrückt, je nachdem, was zuerst erreicht wird. Wenn diese Werte auf die Mindestwerte (60 Sekunden, 1 MB) eingestellt sind, liegt die erwartete Latenz innerhalb von 3 Minuten, sofern die ausgewählten CloudWatch Namespaces über aktive Metrikaktualisierungen verfügen.

In einem CloudWatch Metrik-Stream werden Daten jede Minute gesendet. Die Daten kommen möglicherweise nicht in der richtigen Reihenfolge am endgültigen Ziel an. Alle angegebenen Metriken in den angegebenen Namespaces werden im Metrik-Stream gesendet, mit Ausnahme von Metriken mit einem Zeitstempel, der älter als zwei Tage ist.

Für jede Kombination von Metriknamen und Namespace, die Sie streamen, werden alle Dimensionskombinationen dieses Metriknamens und Namespace gestreamt.

Für Metrik-Streams in Überwachungskonten können Sie wählen, ob Metriken aus den mit diesem Überwachungskonto verknüpften Quellkonten einbezogen werden sollen. Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.

Um Metrik-Streams zu erstellen und zu verwalten, müssen Sie mit einem Konto angemeldet sein, das über die CloudWatchFullAccessRichtlinie und die `iam:PassRole` entsprechende Berechtigung verfügt, oder mit einem Konto, das über die folgende Liste von Berechtigungen verfügt:

- `iam:PassRole`
- `cloudwatch:PutMetricStream`
- `cloudwatch:DeleteMetricStream`
- `cloudwatch:GetMetricStream`
- `cloudwatch:ListMetricStreams`
- `cloudwatch:StartMetricStreams`
- `cloudwatch:StopMetricStreams`

Wenn Sie die für Metrik-Streams erforderliche IAM-Rolle CloudWatch eingerichtet haben möchten, benötigen Sie auch die `iam:PutRolePolicy` Berechtigungen `iam:CreateRole` und.

> **Important**
>
> Ein Benutzer mit der `cloudwatch:PutMetricStream` hat Zugriff auf die CloudWatch Metrikdaten, die gestreamt werden, auch wenn er nicht über die `cloudwatch:GetMetricData` entsprechende Berechtigung verfügt.

Themen

Benutzerdefiniertes Setup mit Firehose

Verwenden Sie diese Methode, um einen Metrik-Stream zu erstellen und ihn an einen Amazon Data Firehose-Lieferstream weiterzuleiten, der Ihre CloudWatch Metriken dorthin weiterleitet, wo Sie sie haben möchten. Sie können sie an einen Data Lake wie Amazon S3 oder an jedes Ziel oder Endgerät streamen, das von Firehose unterstützt wird, einschließlich Drittanbietern.

Die Formate JSON, OpenTelemetry 1.0.0 und OpenTelemetry 0.7.0 werden nativ unterstützt, oder Sie können Transformationen in Ihrem Firehose-Lieferstream konfigurieren, um die Daten in ein anderes Format wie Parquet zu konvertieren. Mit einem Metrik-Stream können Sie die Überwachungsdaten kontinuierlich aktualisieren oder diese CloudWatch Metrikdaten mit

Abrechnungs- und Leistungsdaten kombinieren, um umfangreiche Datensätze zu erstellen. Sie können dann Tools wie Amazon Athena verwenden, um eine Erkenntnis in die Kostenoptimierung, Ressourcenleistung und Ressourcenauslastung zu erhalten.

Sie können die CloudWatch Konsole, die, oder die AWS CLI verwenden AWS CloudFormation, AWS Cloud Development Kit (AWS CDK) um einen Metrik-Stream einzurichten.

Der Firehose-Lieferstream, den Sie für Ihren Metrik-Stream verwenden, muss sich in demselben Konto und in derselben Region befinden, in der Sie den Metrik-Stream eingerichtet haben. Um regionsübergreifende Funktionen zu erreichen, können Sie den Firehose-Lieferstream so konfigurieren, dass er zu einem Endziel gestreamt wird, das sich in einem anderen Konto oder einer anderen Region befindet.

CloudWatch Konsole

In diesem Abschnitt wird beschrieben, wie Sie mit der CloudWatch Konsole einen Metrik-Stream mit Firehose einrichten.

So richten Sie einen benutzerdefinierten Metrik-Stream mit Firehose ein

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken), Streams (Streams) aus. Wählen Sie anschließend Create metric stream (Metrikstream erstellen) aus.
3. (Optional) Wenn Sie mit einem Konto angemeldet sind, das als Überwachungskonto für CloudWatch kontenübergreifende Observability eingerichtet ist, können Sie wählen, ob Metriken von verknüpften Quellkonten in diesen Metrik-Stream aufgenommen werden sollen. Um Metriken aus Quellkonten einzubeziehen, wählen Sie Include source account metrics (Metriken des Quellkontos einbeziehen).
4. Wählen Sie Benutzerdefiniertes Setup mit Firehose.
5. Wählen Sie für Select your Kinesis Data Firehose stream den Firehose Delivery Stream aus, den Sie verwenden möchten. Es muss sich um dasselbe Konto handeln. Das Standardformat für diese Option ist OpenTelemetry 0.7.0, aber Sie können das Format später in diesem Verfahren ändern.

   Wählen Sie dann unter Wählen Sie Ihren Firehose-Lieferstream aus, den Sie verwenden möchten.
6. (Optional) Sie können „Bestehende Servicerolle auswählen“ wählen, um eine bestehende IAM-Rolle zu verwenden, anstatt eine neue für CloudWatch Sie erstellen zu müssen.

7. (Optional) Um das Ausgabeformat vom Standardformat für Ihr Szenario zu ändern, wählen Sie Ausgabeformat ändern. Die unterstützten Formate sind JSON, OpenTelemetry 1.0.0 und OpenTelemetry 0.7.0.
8. Wählen Sie für Metriken, die gestreamt werden sollen, entweder Alle Metriken oder Metriken auswählen aus.

   Wenn Sie Alle Metriken auswählen, werden alle Metriken aus diesem Konto in den Stream aufgenommen.

   Überlegen Sie sorgfältig, ob alle Metriken gestreamt werden sollen, denn je mehr Metriken Sie streamen, desto höher sind Ihre Metrik-Streamgebühren.

   Wenn Sie Metriken auswählen wählen, führen Sie einen der folgenden Schritte aus:

   - Um die meisten Metrik-Namespaces zu streamen, wählen Sie Ausschließen und wählen Sie die Namespaces oder Metriken aus, die ausgeschlossen werden sollen. Wenn Sie in Exclude einen Namespace angeben, können Sie optional einige spezifische Metriken aus diesem Namespace auswählen, die ausgeschlossen werden sollen. Wenn Sie einen Namespace ausschließen, dann aber keine Metriken in diesem Namespace auswählen, werden alle Metriken aus diesem Namespace ausgeschlossen.
   - Um nur einige Metrik-Namespaces oder Metriken in den Metrik-Stream aufzunehmen, wählen Sie Include aus und wählen Sie dann die Namespaces oder Metriken aus, die eingeschlossen werden sollen. Wenn Sie sich dafür entscheiden, einen Namespace einzubeziehen, dann aber keine Metriken in diesem Namespace auswählen, sind alle Metriken aus diesem Namespace enthalten.
9. (Optional) Um zusätzliche Statistiken für einige dieser Metriken als Minimum, Maximum und Summe zu streamen SampleCount, wählen Sie Zusätzliche Statistiken hinzufügen aus. Wählen Sie entweder Add recommended metrics (Empfohlene Metriken hinzufügen) aus, um einige häufig verwendete Statistiken hinzuzufügen, oder wählen Sie manuell den Namespace und den Metriknamen aus, für den Sie zusätzliche Statistiken streamen möchten. Wählen Sie anschließend die zusätzlichen Statistiken zum Streamen aus.

   Anschließend wählen Sie eine weitere Gruppe von Metriken aus. Um eine andere Auswahl zusätzlicher Statistiken zu streamen, wählen Sie Add additional statistics (Zusätzliche Statistiken hinzufügen) aus. Jede Metrik kann bis zu 20 zusätzliche Statistiken enthalten, und bis zu 100 Metriken innerhalb eines Metrikstreams können zusätzliche Statistiken enthalten.

Durch das Streamen zusätzlicher Statistiken entstehen mehr Gebühren. Weitere Informationen finden Sie unter Statistiken, die gestreamt werden können.

Definitionen der zusätzlichen Statistiken finden Sie unter CloudWatch Definitionen von Statistiken.

10. (Optional) Passen Sie den Namen des neuen Metrik-Streams unter Metrik-Stream-Name an.
11. Wählen Sie Metrikstream erstellen aus.

AWS CLI oder AWS API

Gehen Sie wie folgt vor, um einen CloudWatch Metrik-Stream zu erstellen.

Um die AWS CLIAWS OR-API zu verwenden, um einen Metrik-Stream zu erstellen

1. Wenn Sie zu Amazon S3 streamen, erstellen Sie zuerst den Bucket. Weitere Informationen finden Sie unter Bucket erstellen.
2. Erstellen Sie den Firehose-Lieferstream. Weitere Informationen finden Sie unter Einen Firehose-Stream erstellen.
3. Erstellen Sie eine IAM-Rolle, die das Schreiben CloudWatch in den Firehose-Lieferstream ermöglicht. Weitere Informationen über den Inhalt dieser Rolle finden Sie unter Vertrauen zwischen CloudWatch und Firehose.
4. Verwenden Sie den `aws cloudwatch put-metric-stream` CLI-Befehl oder die `PutMetricStream` API, um den CloudWatch Metrik-Stream zu erstellen.

AWS CloudFormation

Sie können ihn verwenden AWS CloudFormation , um einen Metrik-Stream einzurichten. Weitere Informationen finden Sie unter AWS::CloudWatch::MetricStream.

Wird verwendet, AWS CloudFormation um einen Metrik-Stream zu erstellen

1. Wenn Sie zu Amazon S3 streamen, erstellen Sie zuerst den Bucket. Weitere Informationen finden Sie unter Bucket erstellen.
2. Erstellen Sie den Firehose-Lieferstream. Weitere Informationen finden Sie unter Einen Firehose-Stream erstellen.

3. Erstellen Sie eine IAM-Rolle, die das Schreiben CloudWatch in den Firehose-Lieferstream ermöglicht. Weitere Informationen über den Inhalt dieser Rolle finden Sie unter Vertrauen zwischen CloudWatch und Firehose.
4. Erstellen Sie den Stream in. AWS CloudFormation Weitere Informationen finden Sie unter AWS::CloudWatch::MetricStream.

AWS Cloud Development Kit (AWS CDK)

Sie können ihn verwenden AWS Cloud Development Kit (AWS CDK) , um einen Metrik-Stream einzurichten.

Um den zu verwenden AWS CDK , um einen Metrik-Stream zu erstellen

1. Wenn Sie zu Amazon S3 streamen, erstellen Sie zuerst den Bucket. Weitere Informationen finden Sie unter Bucket erstellen.
2. Erstellen Sie den Firehose-Lieferstream. Weitere Informationen finden Sie unter Amazon Data Firehose Delivery Stream erstellen.
3. Erstellen Sie eine IAM-Rolle, die das Schreiben CloudWatch in den Firehose-Lieferstream ermöglicht. Weitere Informationen über den Inhalt dieser Rolle finden Sie unter Vertrauen zwischen CloudWatch und Firehose.
4. Erstellen Sie den Metrik-Stream. Die Metrik-Stream-Ressource ist AWS CDK als Level-1-Konstrukt (L1) mit dem Namen verfügbar. `CfnMetricStream` Weitere Informationen finden Sie unter Verwenden von L1-Konstrukten.

Verwenden von Quick Setup für Amazon S3

Die Methode Quick S3 Setup eignet sich gut, wenn Sie schnell einen Stream zu Amazon S3 einrichten möchten und keine Formatierungstransformation benötigen, die über die unterstützten Formate JSON, OpenTelemetry 1.0.0 und OpenTelemetry 0.7.0 hinausgeht. CloudWatch erstellt alle erforderlichen Ressourcen, einschließlich des Firehose-Lieferstreams und der erforderlichen IAM-Rollen. Das Standardformat für diese Option ist JSON, aber Sie können das Format ändern, während Sie den Stream einrichten.

Wenn Sie möchten, dass das endgültige Format Parquet oder Optimized Row Columnar (ORC) sein soll, sollten Sie stattdessen die Schritte unter Benutzerdefiniertes Setup mit Firehose befolgen.

CloudWatch Konsole

In diesem Abschnitt wird beschrieben, wie Sie mit der CloudWatch Konsole einen Amazon S3 S3-Metrik-Stream mithilfe von Quick S3 Setup einrichten.

So richten Sie einen Metrik-Stream mit S3 Quick Setup ein

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken), Streams (Streams) aus. Wählen Sie anschließend Create metric stream (Metrikstream erstellen) aus.
3. (Optional) Wenn Sie mit einem Konto angemeldet sind, das als Überwachungskonto für CloudWatch kontenübergreifende Observability eingerichtet ist, können Sie wählen, ob Metriken von verknüpften Quellkonten in diesen Metrik-Stream aufgenommen werden sollen. Um Metriken aus Quellkonten einzubeziehen, wählen Sie Include source account metrics (Metriken des Quellkontos einbeziehen).
4. Wählen Sie S3 Quick Setup aus. CloudWatch erstellt alle erforderlichen Ressourcen, einschließlich des Firehose-Lieferstreams und der erforderlichen IAM-Rollen. Das Standardformat für diese Option ist JSON, aber Sie können das Format später in diesem Verfahren ändern.
5. (Optional) Wählen Sie „Vorhandene Ressourcen auswählen“, um einen vorhandenen S3-Bucket oder bestehende IAM-Rollen zu verwenden, anstatt neue für CloudWatch Sie erstellen zu müssen.
6. (Optional) Um das Ausgabeformat vom Standardformat für Ihr Szenario zu ändern, wählen Sie Ausgabeformat ändern. Die unterstützten Formate sind JSON, OpenTelemetry 1.0.0 und OpenTelemetry 0.7.0.
7. Wählen Sie für Metriken, die gestreamt werden sollen, entweder Alle Metriken oder Metriken auswählen aus.

   Wenn Sie Alle Metriken auswählen, werden alle Metriken aus diesem Konto in den Stream aufgenommen.

   Überlegen Sie sorgfältig, ob alle Metriken gestreamt werden sollen, denn je mehr Metriken Sie streamen, desto höher sind Ihre Metrik-Streamgebühren.

   Wenn Sie Metriken auswählen wählen, führen Sie einen der folgenden Schritte aus:

   - Um die meisten Metrik-Namespaces zu streamen, wählen Sie Ausschließen und wählen Sie die Namespaces oder Metriken aus, die ausgeschlossen werden sollen. Wenn Sie

in Exclude einen Namespace angeben, können Sie optional einige spezifische Metriken aus diesem Namespace auswählen, die ausgeschlossen werden sollen. Wenn Sie einen Namespace ausschließen, dann aber keine Metriken in diesem Namespace auswählen, werden alle Metriken aus diesem Namespace ausgeschlossen.

- Um nur einige Metrik-Namespaces oder Metriken in den Metrik-Stream aufzunehmen, wählen Sie Include aus und wählen Sie dann die Namespaces oder Metriken aus, die eingeschlossen werden sollen. Wenn Sie sich dafür entscheiden, einen Namespace einzubeziehen, dann aber keine Metriken in diesem Namespace auswählen, sind alle Metriken aus diesem Namespace enthalten.

8. (Optional) Um zusätzliche Statistiken für einige dieser Metriken als Minimum, Maximum und Summe zu streamen SampleCount, wählen Sie Zusätzliche Statistiken hinzufügen aus. Wählen Sie entweder Add recommended metrics (Empfohlene Metriken hinzufügen) aus, um einige häufig verwendete Statistiken hinzuzufügen, oder wählen Sie manuell den Namespace und den Metriknamen aus, für den Sie zusätzliche Statistiken streamen möchten. Wählen Sie anschließend die zusätzlichen Statistiken zum Streamen aus.

   Anschließend wählen Sie eine weitere Gruppe von Metriken aus. Um eine andere Auswahl zusätzlicher Statistiken zu streamen, wählen Sie Add additional statistics (Zusätzliche Statistiken hinzufügen) aus. Jede Metrik kann bis zu 20 zusätzliche Statistiken enthalten, und bis zu 100 Metriken innerhalb eines Metrikstreams können zusätzliche Statistiken enthalten.

   Durch das Streamen zusätzlicher Statistiken entstehen mehr Gebühren. Weitere Informationen finden Sie unter Statistiken, die gestreamt werden können.

   Definitionen der zusätzlichen Statistiken finden Sie unter CloudWatch Definitionen von Statistiken.
9. (Optional) Passen Sie den Namen des neuen Metrik-Streams unter Metrik-Stream-Name an.
10. Wählen Sie Metrikstream erstellen aus.

Quick Setup für Partner

CloudWatch bietet eine schnelle Einrichtung für die folgenden Drittanbieter. Um diesen Workflow verwenden zu können, müssen Sie nur eine Ziel-URL und einen API-Schlüssel für Ihr Ziel angeben. CloudWatch kümmert sich um den Rest der Einrichtung, einschließlich der Erstellung des Firehose-Lieferdatenstroms und der erforderlichen IAM-Rollen.

> **Important**
>
> Bevor Sie Quick Setup für Partner verwenden, um einen Metrik-Stream zu erstellen, empfehlen wir Ihnen dringend, die Dokumentation dieses Partners zu lesen, die in der folgenden Liste verlinkt ist.

- Datadog
- Dynatrace
- New Relic
- Splunk Observability Cloud
- SumoLogic

Wenn Sie einen Metrik-Stream für einen dieser Partner einrichten, wird der Stream mit den in den folgenden Abschnitten aufgeführten Standardeinstellungen erstellt.

Themen

Mithilfe von Quick Setup für Partner einen Metrik-Stream einrichten

CloudWatch bietet eine schnelle Einrichtungsoption für einige Drittanbieter. Bevor Sie mit den Schritten in diesem Abschnitt ausführen, benötigen Sie bestimmte Informationen für den Partner. Diese Informationen können eine Ziel-URL und/oder einen API-Schlüssel für Ihr Partnerziel enthalten. Sie sollten auch die Dokumentation auf der im vorherigen Abschnitt verlinkten Website des Partners sowie die in den folgenden Abschnitten aufgeführten Standardeinstellungen für diesen Partner lesen.

Um an ein Drittanbieter-Ziel zu streamen, das von Quick Setup nicht unterstützt wird, können Sie den Anweisungen unter Folgen Sie den Anweisungen unter Benutzerdefiniertes Setup mit Firehose

So richten Sie einen Stream mit Firehose ein, folgen und diese Metriken dann von Firehose an das endgültige Ziel senden.

So verwenden Sie Quick Setup für Partner, um einen Metrik-Stream an einen Drittanbieter zu erstellen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken), Streams (Streams) aus. Wählen Sie anschließend Create metric stream (Metrikstream erstellen) aus.
3. (Optional) Wenn Sie mit einem Konto angemeldet sind, das als Überwachungskonto für CloudWatch kontenübergreifende Observability eingerichtet ist, können Sie wählen, ob Metriken von verknüpften Quellkonten in diesen Metrik-Stream aufgenommen werden sollen. Um Metriken aus Quellkonten einzubeziehen, wählen Sie Include source account metrics (Metriken des Quellkontos einbeziehen).
4. Wählen Sie Schnelles Amazon Web Services Services-Partner-Setup
5. Klicken Sie auf den Namen des Partners, an den Sie Metriken streamen möchten.
6. Geben Sie für Endpunkt-URL die Ziel-URL ein.
7. Geben Sie für Zugriffsschlüssel oder API-Schlüssel den Zugriffsschlüssel für den Partner ein. Nicht alle Partner benötigen einen Zugriffsschlüssel.
8. Wählen Sie für Metriken, die gestreamt werden sollen, entweder Alle Metriken oder Metriken auswählen aus.

   Wenn Sie Alle Metriken auswählen, werden alle Metriken aus diesem Konto in den Stream aufgenommen.

   Überlegen Sie sorgfältig, ob alle Metriken gestreamt werden sollen, denn je mehr Metriken Sie streamen, desto höher sind Ihre Metrik-Streamgebühren.

   Wenn Sie Metriken auswählen wählen, führen Sie einen der folgenden Schritte aus:

   - Um die meisten Metrik-Namespaces zu streamen, wählen Sie Ausschließen und wählen Sie die Namespaces oder Metriken aus, die ausgeschlossen werden sollen. Wenn Sie in Exclude einen Namespace angeben, können Sie optional einige spezifische Metriken aus diesem Namespace auswählen, die ausgeschlossen werden sollen. Wenn Sie einen Namespace ausschließen, dann aber keine Metriken in diesem Namespace auswählen, werden alle Metriken aus diesem Namespace ausgeschlossen.

- Um nur einige Metrik-Namespaces oder Metriken in den Metrik-Stream aufzunehmen, wählen Sie Include aus und wählen Sie dann die Namespaces oder Metriken aus, die eingeschlossen werden sollen. Wenn Sie sich dafür entscheiden, einen Namespace einzubeziehen, dann aber keine Metriken in diesem Namespace auswählen, sind alle Metriken aus diesem Namespace enthalten.

9. (Optional) Um zusätzliche Statistiken für einige dieser Metriken als Minimum, Maximum und Summe zu streamen SampleCount, wählen Sie Zusätzliche Statistiken hinzufügen aus. Wählen Sie entweder Add recommended metrics (Empfohlene Metriken hinzufügen) aus, um einige häufig verwendete Statistiken hinzuzufügen, oder wählen Sie manuell den Namespace und den Metriknamen aus, für den Sie zusätzliche Statistiken streamen möchten. Wählen Sie anschließend die zusätzlichen Statistiken zum Streamen aus.

   Anschließend wählen Sie eine weitere Gruppe von Metriken aus. Um eine andere Auswahl zusätzlicher Statistiken zu streamen, wählen Sie Add additional statistics (Zusätzliche Statistiken hinzufügen) aus. Jede Metrik kann bis zu 20 zusätzliche Statistiken enthalten, und bis zu 100 Metriken innerhalb eines Metrikstreams können zusätzliche Statistiken enthalten.

   Durch das Streamen zusätzlicher Statistiken entstehen mehr Gebühren. Weitere Informationen finden Sie unter Statistiken, die gestreamt werden können.

   Definitionen der zusätzlichen Statistiken finden Sie unter CloudWatch Definitionen von Statistiken.
10. (Optional) Passen Sie den Namen des neuen Metrik-Streams unter Metrik-Stream-Name an.
11. Wählen Sie Metrikstream erstellen aus.

Standardwerte für den Datadog-Stream

Streams an Datadog von Quick Setup für Partner verwenden die folgenden Standardeinstellungen:

- Ausgabeformat: OpenTelemetry 0.7.0
- Firehose-Stream-Inhaltskodierung GZIP
- Firehose-Stream-Pufferoptionen Intervall von 60 Sekunden, Größe von 4 MB/s
- Firehose-Stream-Wiederholungsoption Dauer von 60 Sekunden

Wenn Sie Quick Setup für Partner verwenden, um einen Metrik-Stream zu Datadog zu erstellen und Sie bestimmte Metriken streamen, enthalten diese Metriken standardmäßig einige zusätzliche

Statistiken. Für das Streaming zusätzlicher Statistiken können zusätzliche Gebühren anfallen. Weitere Informationen über Statistiken und ihre Kosten finden Sie unter Statistiken, die gestreamt werden können.

Die folgende Liste zeigt die Metriken, für die standardmäßig zusätzliche Statistiken gestreamt werden, falls Sie sich dafür entscheiden, diese Metriken zu streamen. Sie können die Auswahl dieser zusätzlichen Statistiken aufheben, bevor Sie den Stream starten.

- **`Duration`** in **`AWS/Lambda`**: p50, p80, p95, p99, p99.9
- **`PostRuntimeExtensionDuration`** in **`AWS/Lambda`**: p50, p99
- **`FirstByteLatency`** und **`TotalRequestLatency`** in **`AWS/S3`**: p50, p90, p95, p99, p99.9
- **`ResponseLatency`** in **`AWS/Polly`** und **`TargetResponseTime`** in AWS/ApplicationELB: p50, p90, p95, p99
- **`Latency`** und **`IntegrationLatency`** in **`AWS/ApiGateway`**: p90, p95, p99
- **`Latency`** und **`TargetResponseTime`** in **`AWS/ELB`**: p95, p99
- **`RequestLatency`** in **`AWS/AppRunner`**: p50, p95, p99
- **`ActivityTime`**, **`ExecutionTime`**, **`LambdaFunctionRunTime`**, **`LambdaFunctionScheduleTime`**, **`LambdaFunctionTime`**, **`ActivityRunTime`** und **`ActivityScheduleTime`** in **`AWS/States`**: p95, p99
- **`EncoderBitRate`**, **`ConfiguredBitRate`** und **`ConfiguredBitRateAvailable`** in **`AWS/MediaLive`**: p90
- **`Latency`** in **`AWS/AppSync`**: p90

Standardwerte für den Dynatrace-Stream

Streams an Dynatrace von Quick Setup für Partner verwenden die folgenden Standardeinstellungen:

- Ausgabeformat: 0.7.0 OpenTelemetry
- Firehose-Stream-Inhaltskodierung GZIP
- Firehose-Stream-Pufferoptionen Intervall von 60 Sekunden, Größe von 5 MB/s
- Firehose-Stream-Wiederholungsoption Dauer von 600 Sekunden

Standardwerte für den New-Relic-Stream

Streams an New Relic von Quick Setup für Partner verwenden die folgenden Standardeinstellungen:

- Ausgabeformat: 0.7.0 OpenTelemetry
- Firehose-Stream-Inhaltskodierung GZIP
- Firehose-Stream-Pufferoptionen Intervall von 60 Sekunden, Größe von 1 MB
- Firehose-Stream-Wiederholungsoption Dauer von 60 Sekunden

Die Standardeinstellungen für den Splunk-Observability-Cloud-Stream

Streams an Splunk Observability Cloud von Quick Setup für Partner verwenden die folgenden Standardeinstellungen:

- Ausgabeformat: 0.7.0 OpenTelemetry
- Firehose-Stream-Inhaltskodierung GZIP
- Firehose-Stream-Pufferoptionen Intervall von 60 Sekunden, Größe von 1 MB
- Firehose-Stream-Wiederholungsoption Dauer von 300 Sekunden

Standardwerte für den Sumo-Logic-Stream

Streams an Sumo Logic von Quick Setup für Partner verwenden die folgenden Standardeinstellungen:

- Ausgabeformat: 0.7.0 OpenTelemetry
- Firehose-Stream-Inhaltskodierung GZIP
- Firehose-Stream-Pufferoptionen Intervall von 60 Sekunden, Größe von 1 MB
- Firehose-Stream-Wiederholungsoption Dauer von 60 Sekunden

Statistiken, die gestreamt werden können

Metrikstreams beinhalten immer die folgenden Statistiken: `Minimum`, `Maximum`, `SampleCount` und `Sum`. Sie können auch die folgenden zusätzlichen Statistiken in einen Metrikstream aufnehmen. Sie entscheiden für jede Metrik individuell. Weitere Informationen zu diesen Statistiken finden Sie unter CloudWatch Definitionen von Statistiken.

- Perzentilwerte wie p95 oder p99 (für Streams mit JSON oder Format) OpenTelemetry
- Getrimmter Mittelwert (nur für Streams im JSON-Format)

- Winsorized-Mittelwert (nur für Streams im JSON-Format)
- Getrimmte Anzahl (nur für Streams im JSON-Format)
- Getrimmte Summe (nur für Streams im JSON-Format)
- Perzentilrang (nur für Streams im JSON-Format)
- Interquartil-Mittelwert (nur für Streams im JSON-Format)

Durch das Streamen zusätzlicher Statistiken entstehen zusätzliche Gebühren. Das Streaming zwischen einer und fünf dieser zusätzlichen Statistiken für eine bestimmte Metrik wird als zusätzliche Metrik-Aktualisierung in Rechnung gestellt. Danach wird jeder weitere Satz von bis zu fünf dieser Statistiken als weitere Metrik-Aktualisierung in Rechnung gestellt.

Angenommen, Sie streamen für eine Metrik die folgenden sechs zusätzlichen Statistiken: p95, p99, p99,9, Getrimmter Mittelwert, Winsorized-Mittelwert und Getrimmte Summe. Jede Aktualisierung dieser Metrik wird in Form von drei Metrikaktualisierungen abgerechnet: einmal für die Metrik-Aktualisierung, die die Standardstatistik enthält, einmal für die ersten fünf zusätzlichen Statistiken und einmal für die sechste zusätzliche Statistik. Das Hinzufügen von bis zu vier weiteren zusätzlichen Statistiken für insgesamt zehn würde die Abrechnung nicht erhöhen, aber eine elfte zusätzliche Statistik würde zu einer Erhöhung führen.

Wenn Sie eine Kombination aus Metriknamen und Namespace angeben, um zusätzliche Statistiken zu streamen, werden alle Dimensionskombinationen dieses Metriknamens und Namespace mit den zusätzlichen Statistiken gestreamt.

CloudWatch metric streams veröffentlicht eine neue Metrik`TotalMetricUpdate`, die die Basisanzahl der Metrikaktualisierungen sowie zusätzliche Metrikaktualisierungen, die durch das Streaming zusätzlicher Statistiken entstehen, widerspiegelt. Weitere Informationen finden Sie unter Überwachen Sie Ihre Metrik-Streams mit CloudWatch Metriken.

Weitere Informationen finden Sie unter CloudWatch Amazon-Preise.

> **Note**
>
> Einige Metriken unterstützen keine Perzentile. Perzentilstatistiken für diese Metriken sind vom Stream ausgeschlossen und verursachen keine Metrikstream-Gebühren. Ein Beispiel für diese Statistiken, die keine Perzentile unterstützen, sind einige Metriken im `AWS/ECS`-Namespace.

Die zusätzlichen Statistiken, die Sie konfigurieren, werden nur gestreamt, wenn sie mit den Filtern für den Stream übereinstimmen. Wenn Sie beispielsweise einen Stream erstellen, der nur `EC2` und `RDS` in den Include-Filtern und dann Ihre Statistikkonfigurationslisten `EC2` und `Lambda` enthält, dann enthält der Stream `EC2`-Metriken mit zusätzlichen Statistiken, `RDS`-Metriken mit nur der Standardstatistik und keine `Lambda`-Statistiken.

Betrieb und Wartung von Metrik-Streams

Metrikstreams befinden sich immer in einem von zwei Zuständen, Wird ausgeführt oder Angehalten.

- Wird ausgeführt – Der Metrik-Stream wird ordnungsgemäß ausgeführt. Möglicherweise werden aufgrund der Filter im Stream keine Metrikdaten zum Ziel gestreamt.
- Angehalten – Der Metrik-Stream wurde explizit von jemandem angehalten und nicht wegen eines Fehlers. Es kann nützlich sein, den Stream zu stoppen, um das Streamen von Daten vorübergehend anzuhalten, ohne den Stream zu löschen.

Wenn Sie einen Metrik-Stream beenden und neu starten, werden die Metrikdaten, die veröffentlicht wurden, CloudWatch während der Metrik-Stream gestoppt wurde, nicht wieder in den Metrik-Stream übernommen.

Wenn Sie das Ausgabeformat eines Metrik-Streams ändern, wird in bestimmten Fällen möglicherweise eine kleine Menge an Metrikdaten angezeigt, die sowohl im alten als auch im neuen Format an das Ziel geschrieben wurden. Um diese Situation zu vermeiden, können Sie einen neuen Firehose-Lieferstream mit derselben Konfiguration wie Ihren aktuellen erstellen, dann zum neuen Firehose-Lieferstream wechseln und gleichzeitig das Ausgabeformat ändern. Auf diese Weise werden die Kinesis Datensätze mit unterschiedlichem Ausgabeformat in Amazon S3 in separaten Objekten gespeichert. Später können Sie den Verkehr zurück zum ursprünglichen Firehose-Lieferstream leiten und den zweiten Lieferstream löschen.

So zeigen Sie Ihre Metrik-Streams an, bearbeiten, stoppen und starten Sie sie

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken), Streams (Streams) aus.

   Die Liste der Streams wird angezeigt, und die Spalte Status zeigt an, ob jeder Stream ausgeführt oder angehalten wird.
3. Um einen Metrik-Stream zu stoppen oder zu starten, wählen Sie den Stream aus und wählen Sie Stopp oder Start.

4. Um die Details zu einem Metrikstream anzuzeigen, wählen Sie den Stream aus und wählen Sie Details anzeigen.
5. Um das Ausgabeformat, die Filter, den Firehose-Zielstream oder die Rollen des Streams zu ändern, wählen Sie Bearbeiten und nehmen Sie die gewünschten Änderungen vor.

   Wenn Sie die Filter ändern, können während des Übergangs Lücken in den Metrikdaten auftreten.

Überwachen Sie Ihre Metrik-Streams mit CloudWatch Metriken

Metrische Streams geben CloudWatch Metriken über ihren Zustand und Betrieb im `AWS/CloudWatch/MetricStreams` Namespace aus. Die folgenden Metriken werden ausgegeben. Beide Metriken werden mit einer `MetricStreamName`-Dimension und ohne Dimension ausgegeben. Sie können die Metriken ohne Dimensionen verwenden, um aggregierte Metriken für alle Ihre Metrik-Streams anzuzeigen. Sie können die Metriken mit der `MetricStreamName`-Dimension verwenden, um die Metriken nur für diesen Metrikstream anzuzeigen.

Für diese Metriken werden Werte nur für Metrikstreams ausgegeben, die sich im Status Running (Wird ausgeführt) befinden.

Metrik	Beschreibung
`MetricUpdate`	Die Anzahl der Metrikaktualisierungen, die an den Metrik-Stream gesendet werden. Wenn während eines Zeitraums keine Metrikakt ualisierungen gestreamt werden, wird diese Metrik während dieses Zeitraums nicht ausgegeben. Wenn Sie den Metrik-Stream stoppen, wird diese Metrik nicht mehr ausgegeben, bis der Metrik-Stream erneut gestartet wird. Gültige Statistik: `Sum` Einheiten: keine
`TotalMetr icUpdate`	Dies wird auf der Grundlage zusätzlicher Statistiken, die gestreamt werden, als Zahl MetricUpdate + berechnet. Für jede Kombination aus eindeutigem Namespace und Metriknamen fügt das Streamen von 1-5 zusätzlichen Statistiken 1 zu `TotalMetr`

Metrik	Beschreibung
	`icUpdate` hinzu. Das Streaming von 6-10 zusätzlichen Statistiken fügt 2 zu `TotalMetricUpdate` hinzu usw. Gültige Statistik: `Sum` Einheiten: keine
`PublishErrorRate`	Die Anzahl der nicht behebbaren Fehler, die beim Einfügen von Daten in den Firehose-Lieferstream auftreten. Wenn während eines Zeitraums keine Fehler auftreten, wird diese Metrik während dieses Zeitraums nicht ausgegeben. Wenn Sie den Metrik-Stream stoppen, wird diese Metrik nicht mehr ausgegeben, bis der Metrik-Stream erneut gestartet wird. Gültige Statistik: `Average`, um die Rate der Metrikaktualisierungen anzuzeigen, die nicht geschrieben werden können. Dieser Wert muss zwischen 0,0 und 1,0 liegen. Einheiten: keine

Vertrauen zwischen CloudWatch und Firehose

Der Firehose-Lieferstream muss CloudWatch über eine IAM-Rolle, die über Schreibberechtigungen für Firehose verfügt, vertrauenswürdig sein. Diese Berechtigungen können auf den einzelnen Firehose-Lieferstream beschränkt werden, den der CloudWatch Metrik-Stream verwendet. Die IAM-Rolle muss dem Service-Prinzipal `streams.metrics.cloudwatch.amazonaws.com` vertrauen.

Wenn Sie die CloudWatch Konsole verwenden, um einen Metrik-Stream zu CloudWatch erstellen, können Sie die Rolle mit den richtigen Berechtigungen erstellen lassen. Wenn Sie eine andere Methode verwenden, um einen Metrik-Stream zu erstellen oder die IAM-Rolle selbst erstellen möchten, muss diese die folgende Berechtigungsrichtlinie und Vertrauensrichtlinie enthalten.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
```

```
                "firehose:PutRecord",
                "firehose:PutRecordBatch"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:firehose:region:account-id:deliverystream/*"
        }
    ]
}
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "streams.metrics.cloudwatch.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Metrikdaten werden im Namen der Quelle CloudWatch , der die Metrik-Stream-Ressource gehört, an den Firehose-Ziel-Lieferstream gestreamt.

Ausgabeformate für Metrik-Streams

Die Daten in einem CloudWatch Metrik-Stream können im JSON-Format oder im folgenden Format vorliegen. OpenTelemetry Derzeit werden sowohl die Formate OpenTelemetry 1.0.0 als auch 0.7.0 unterstützt.

Inhalt

- Wie analysiert man 0.7.0-Nachrichten OpenTelemetry

JSON-Format

In einem CloudWatch Metrik-Stream, der das JSON-Format verwendet, enthält jeder Firehose-Datensatz mehrere JSON-Objekte, die durch ein Zeilenumbruchzeichen (\n) getrennt sind. Jedes Objekt enthält einen einzelnen Datenpunkt einer einzelnen Metrik.

Das verwendete JSON-Format ist vollständig kompatibel mit AWS Glue und mit Amazon Athena. Wenn Sie einen Firehose-Lieferstream und eine korrekt formatierte AWS Glue Tabelle haben, kann das Format automatisch in das Parquet-Format oder das Optimized Row Columnar (ORC) -Format umgewandelt werden, bevor es in S3 gespeichert wird. Weitere Informationen zum Transformieren des Formats finden Sie unter Konvertieren Ihres Eingabedatensatzformats in Firehose. Weitere Informationen zum richtigen Format für finden Sie AWS Glue unterWelches AWS Glue -Schema sollte ich für das JSON-Ausgabeformat verwenden?.

Im JSON-Format sind die gültigen Werte für `unit` dieselben wie für den Wert von `unit` in der `MetricDatum`-API-Struktur. Weitere Informationen finden Sie unter MetricDatum. Der Wert für das `timestamp`-Feld ist in Epochen-Millisekunden angegeben, z. B. `1616004674229`.

Nachfolgend sehen Sie ein Beispiel des Formats. In diesem Beispiel wird der JSON zum einfachen Lesen formatiert, in der Praxis befindet sich das gesamte Format in einer einzelnen Zeile.

```
{
    "metric_stream_name": "MyMetricStream",
    "account_id": "1234567890",
    "region": "us-east-1",
    "namespace": "AWS/EC2",
    "metric_name": "DiskWriteOps",
    "dimensions": {
        "InstanceId": "i-123456789012"
    },
    "timestamp": 1611929698000,
    "value": {
        "count": 3.0,
        "sum": 20.0,
        "max": 18.0,
        "min": 0.0,
        "p99": 17.56,
        "p99.9": 17.8764,
```

```
        "TM(25%:75%)": 16.43
    },
    "unit": "Seconds"
}
```

Welches AWS Glue -Schema sollte ich für das JSON-Ausgabeformat verwenden?

Das Folgende ist ein Beispiel für eine JSON-Darstellung von `StorageDescriptor` für eine AWS Glue Tabelle, die dann von Firehose verwendet würde. Weitere Informationen zu finden Sie `StorageDescriptor` unter StorageDescriptor.

```
{
  "Columns": [
    {
      "Name": "metric_stream_name",
      "Type": "string"
    },
    {
      "Name": "account_id",
      "Type": "string"
    },
    {
      "Name": "region",
      "Type": "string"
    },
    {
      "Name": "namespace",
      "Type": "string"
    },
    {
      "Name": "metric_name",
      "Type": "string"
    },
    {
      "Name": "timestamp",
      "Type": "timestamp"
    },
    {
      "Name": "dimensions",
      "Type": "map<string,string>"
    },
    {
      "Name": "value",
```

```
      "Type":
 "struct<min:double,max:double,count:double,sum:double,p99:double,p99.9:double>"
    },
    {
      "Name": "unit",
      "Type": "string"
    }
  ],
  "Location": "s3://my-s3-bucket/",
  "InputFormat": "org.apache.hadoop.mapred.TextInputFormat",
  "OutputFormat": "org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat",
  "SerdeInfo": {
    "SerializationLibrary": "org.apache.hive.hcatalog.data.JsonSerDe"
  },
  "Parameters": {
    "classification": "json"
  }
}
```

Das obige Beispiel bezieht sich auf Daten, die auf Amazon S3 im JSON-Format geschrieben wurden. Ersetzen Sie die Werte in den folgenden Feldern durch die angegebenen Werte, um die Daten im Parquet-Format oder im Optimized Row Columnar (ORC)-Format zu speichern.

- Parquet:
  - Eingabeformat: org.apache.hadoop.hive.ql.io.parquet. MapredParquetInputFormat
  - Ausgabeformat: org.apache.hadoop.hive.ql.io.parquet. MapredParquetOutputFormat
  - SerDeInfo.serializationLib: org.apache.hadoop.hive.ql.io.parquet.serde. ParquetHiveSerDe
  - parameters.classification: parquet
- ORC:
  - Eingabeformat: org.apache.hadoop.hive.ql.io.orc. OrcInputFormat
  - Ausgabeformat: org.apache.hadoop.hive.ql.io.orc. OrcOutputFormat
  - SerDeInfo.serializationLib: org.apache.hadoop.hive.ql.io.orc. OrcSerde
  - parameters.classification: orc

OpenTelemetry 1.0.0-Format

> **Note**
>
> Beim Format OpenTelemetry 1.0.0 werden metrische Attribute als eine Liste von `KeyValue` Objekten kodiert und nicht als der `StringKeyValue` Typ, der im Format 0.7.0 verwendet wird. Für Verbraucher ist dies die einzige wichtige Änderung zwischen den Formaten 0.7.0 und 1.0.0. Ein aus den 0.7.0-Protodateien generierter Parser analysiert keine metrischen Attribute, die im Format 1.0.0 codiert sind. Das Gleiche gilt umgekehrt: Ein aus den 1.0.0-Protodateien generierter Parser analysiert keine im 0.7.0-Format kodierten metrischen Attribute.

OpenTelemetry ist eine Sammlung von Tools, APIs und SDKs. Sie können damit Telemetriedaten (Metriken, Protokolle und Traces) für Analysen instrumentieren, generieren, sammeln und exportieren. OpenTelemetry ist Teil der Cloud Native Computing Foundation. Weitere Informationen finden Sie unter OpenTelemetry.

Informationen zur vollständigen OpenTelemetry 1.0.0-Spezifikation finden Sie unter Release-Version 1.0.0.

Ein Kinesis-Datensatz kann eine oder mehrere `ExportMetricsServiceRequest` OpenTelemetry Datenstrukturen enthalten. Jede Datenstruktur beginnt mit einem Header mit einem `UnsignedVarInt32`, das die Datensatzlänge in Byte angibt. Jede `ExportMetricsServiceRequest` kann Daten von mehreren Metriken gleichzeitig enthalten.

Das Folgende ist eine Zeichenkettendarstellung der Botschaft der `ExportMetricsServiceRequest` OpenTelemetry Datenstruktur. OpenTelemetry serialisiert das Binärprotokoll von Google Protocol Buffers, und dieses ist nicht für Menschen lesbar.

```
resource_metrics {
  resource {
    attributes {
      key: "cloud.provider"
      value {
        string_value: "aws"
      }
    }
    attributes {
      key: "cloud.account.id"
```

```
      value {
        string_value: "123456789012"
      }
    }
    attributes {
      key: "cloud.region"
      value {
        string_value: "us-east-1"
      }
    }
    attributes {
      key: "aws.exporter.arn"
      value {
        string_value: "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/
MyMetricStream"
      }
    }
  }
  scope_metrics {
    metrics {
      name: "amazonaws.com/AWS/DynamoDB/ConsumedReadCapacityUnits"
      unit: "NoneTranslated"
      summary {
        data_points {
          start_time_unix_nano: 60000000000
          time_unix_nano: 120000000000
          count: 1
          sum: 1.0
          quantile_values {
            value: 1.0
          }
          quantile_values {
            quantile: 0.95
            value: 1.0
          }
          quantile_values {
            quantile: 0.99
            value: 1.0
          }
          quantile_values {
            quantile: 1.0
            value: 1.0
          }
          attributes {
```

```
            key: "Namespace"
            value {
              string_value: "AWS/DynamoDB"
            }
          }
          attributes {
            key: "MetricName"
            value {
              string_value: "ConsumedReadCapacityUnits"
            }
          }
          attributes {
            key: "Dimensions"
            value {
              kvlist_value {
                values {
                  key: "TableName"
                  value {
                    string_value: "MyTable"
                  }
                }
              }
            }
          }
        }
        data_points {
          start_time_unix_nano: 70000000000
          time_unix_nano: 130000000000
          count: 2
          sum: 5.0
          quantile_values {
            value: 2.0
          }
          quantile_values {
            quantile: 1.0
            value: 3.0
          }
          attributes {
            key: "Namespace"
            value {
              string_value: "AWS/DynamoDB"
            }
          }
          attributes {
```

```
            key: "MetricName"
            value {
              string_value: "ConsumedReadCapacityUnits"
            }
          }
          attributes {
            key: "Dimensions"
            value {
              kvlist_value {
                values {
                  key: "TableName"
                  value {
                    string_value: "MyTable"
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

Objekt der obersten Ebene zur Serialisierung von Metrikdaten OpenTelemetry

`ExportMetricsServiceRequest`ist der Wrapper der obersten Ebene zum Serialisieren einer Export-Payload. OpenTelemetry Es enthält eine oder mehrere `ResourceMetrics`.

```
message ExportMetricsServiceRequest {
  // An array of ResourceMetrics.
  // For data coming from a single resource this array will typically contain one
  // element. Intermediary nodes (such as OpenTelemetry Collector) that receive
  // data from multiple origins typically batch the data before forwarding further and
  // in that case this array will contain multiple elements.
  repeated opentelemetry.proto.metrics.v1.ResourceMetrics resource_metrics = 1;
}
```

`ResourceMetrics`ist das Objekt der obersten Ebene zur Darstellung von Objekten. MetricData

```
// A collection of ScopeMetrics from a Resource.
message ResourceMetrics {
  reserved 1000;
```

```
  // The resource for the metrics in this message.
  // If this field is not set then no resource info is known.
  opentelemetry.proto.resource.v1.Resource resource = 1;

  // A list of metrics that originate from a resource.
  repeated ScopeMetrics scope_metrics = 2;

  // This schema_url applies to the data in the "resource" field. It does not apply
  // to the data in the "scope_metrics" field which have their own schema_url field.
  string schema_url = 3;
}
```

Ressourcenobjekt

Ein `Resource`-Objekt ist ein Wertpaarobjekt, das Informationen über die Ressource enthält, die die Metriken generiert hat. Bei Metriken, die von AWS erstellt wurden, enthält die Datenstruktur den Amazon-Ressourcennamen (ARN) der Ressource, die sich auf die Metrik bezieht, z. B. eine EC2-Instance oder ein S3 Bucket.

Das `Resource`-Objekt enthält ein Attribut namens `attributes`, das eine Liste von Schlüssel-Wert-Paaren speichert.

- `cloud.account.id` enthält die Konto-ID
- `cloud.region` enthält die Region
- `aws.exporter.arn` enthält den Metrik-Stream-ARN
- `cloud.provider` ist immer `aws`.

```
// Resource information.
message Resource {
  // Set of attributes that describe the resource.
  // Attribute keys MUST be unique (it is not allowed to have more than one
  // attribute with the same key).
  repeated opentelemetry.proto.common.v1.KeyValue attributes = 1;

  // dropped_attributes_count is the number of dropped attributes. If the value is 0,
 then
  // no attributes were dropped.
  uint32 dropped_attributes_count = 2;
}
```

Das Objekt ScopeMetrics

Das `scope`-Feld wird nicht ausgefüllt. Wir füllen nur das Metrikfeld aus, das wir exportieren.

```
// A collection of Metrics produced by an Scope.
message ScopeMetrics {
  // The instrumentation scope information for the metrics in this message.
  // Semantically when InstrumentationScope isn't set, it is equivalent with
  // an empty instrumentation scope name (unknown).
  opentelemetry.proto.common.v1.InstrumentationScope scope = 1;

  // A list of metrics that originate from an instrumentation library.
  repeated Metric metrics = 2;

  // This schema_url applies to all metrics in the "metrics" field.
  string schema_url = 3;
}
```

Metrikobjekt

Das Metrikobjekt enthält einige Metadaten und ein `Summary`-Datenfeld, das eine Liste von `SummaryDataPoint` enthält.

Für Metrik-Streams lauten die Metadaten wie folgt:

- `name` wird `amazonaws.com/`*`metric_namespace`*`/`*`metric_name`* sein.
- `description` wird leer sein
- `unit` wird gefüllt, indem die Einheit des metrischen Datums auf die Variante des einheitlichen Codes für Maßeinheiten, bei der die Groß-/Kleinschreibung beachtet wird, abgebildet wird. Weitere Informationen finden Sie unter Übersetzungen im Format OpenTelemetry 1.0.0 und Der einheitliche Code für Maßeinheiten.
- `type` wird SUMMARY sein.

```
message Metric {
  reserved 4, 6, 8;

  // name of the metric, including its DNS name prefix. It must be unique.
  string name = 1;

  // description of the metric, which can be used in documentation.
```

```
  string description = 2;

  // unit in which the metric value is reported. Follows the format
  // described by http://unitsofmeasure.org/ucum.html.
  string unit = 3;

  // Data determines the aggregation type (if any) of the metric, what is the
  // reported value type for the data points, as well as the relatationship to
  // the time interval over which they are reported.
  oneof data {
    Gauge gauge = 5;
    Sum sum = 7;
    Histogram histogram = 9;
    ExponentialHistogram exponential_histogram = 10;
    Summary summary = 11;
  }
}

message Summary {
  repeated SummaryDataPoint data_points = 1;
}
```

Das SummaryDataPoint Objekt

Das SummaryDataPoint Objekt enthält den Wert eines einzelnen Datenpunkts in einer Zeitreihe in einer DoubleSummary Metrik.

```
// SummaryDataPoint is a single data point in a timeseries that describes the
// time-varying values of a Summary metric.
message SummaryDataPoint {
  reserved 1;

  // The set of key/value pairs that uniquely identify the timeseries from
  // where this point belongs. The list may be empty (may contain 0 elements).
  // Attribute keys MUST be unique (it is not allowed to have more than one
  // attribute with the same key).
  repeated opentelemetry.proto.common.v1.KeyValue attributes = 7;

  // StartTimeUnixNano is optional but strongly encouraged, see the
  // the detailed comments above Metric.
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
```

```
  fixed64 start_time_unix_nano = 2;

  // TimeUnixNano is required, see the detailed comments above Metric.
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  fixed64 time_unix_nano = 3;

  // count is the number of values in the population. Must be non-negative.
  fixed64 count = 4;

  // sum of the values in the population. If count is zero then this field
  // must be zero.
  //
  // Note: Sum should only be filled out when measuring non-negative discrete
  // events, and is assumed to be monotonic over the values of these events.
  // Negative events *can* be recorded, but sum should not be filled out when
  // doing so.  This is specifically to enforce compatibility w/ OpenMetrics,
  // see: https://github.com/OpenObservability/OpenMetrics/blob/main/specification/
OpenMetrics.md#summary
  double sum = 5;

  // Represents the value at a given quantile of a distribution.
  //
  // To record Min and Max values following conventions are used:
  // - The 1.0 quantile is equivalent to the maximum value observed.
  // - The 0.0 quantile is equivalent to the minimum value observed.
  //
  // See the following issue for more context:
  // https://github.com/open-telemetry/opentelemetry-proto/issues/125
  message ValueAtQuantile {
    // The quantile of a distribution. Must be in the interval
    // [0.0, 1.0].
    double quantile = 1;

    // The value at the given quantile of a distribution.
    //
    // Quantile values must NOT be negative.
    double value = 2;
  }

  // (Optional) list of values at different quantiles of the distribution calculated
  // from the current snapshot. The quantiles must be strictly increasing.
  repeated ValueAtQuantile quantile_values = 6;
```

```
  // Flags that apply to this specific data point.  See DataPointFlags
  // for the available flags and their meaning.
  uint32 flags = 8;
}
```

Weitere Informationen finden Sie unter Übersetzungen im Format OpenTelemetry 1.0.0.

Übersetzungen im Format OpenTelemetry 1.0.0

CloudWatch führt einige Transformationen durch, um CloudWatch Daten in OpenTelemetry ein Format zu bringen.

Namespace, Metriknamen und Dimensionen übersetzen

Diese Attribute sind Schlüssel-Wert-Paare, die im Mapping kodiert sind.

- Ein Attribut hat den Schlüssel `Namespace` und sein Wert ist der Namespace der Metrik
- Ein Attribut hat den Schlüssel `MetricName` und sein Wert ist der Name der Metrik
- Ein Paar hat den Schlüssel `Dimensions` und sein Wert ist eine verschachtelte Liste von Schlüssel-Wert-Paaren. Jedes Paar in dieser Liste ist einer CloudWatch metrischen Dimension zugeordnet, wobei der Schlüssel des Paares der Name der Dimension und sein Wert der Wert der Dimension ist.

Dabei werden Durchschnitt, Summe SampleCount, Min und Max übersetzt

Der Zusammenfassungsdatenpunkt ermöglicht CloudWatch den Export all dieser Statistiken unter Verwendung eines Datenpunkts.

- `startTimeUnixNano`enthält die CloudWatch `startTime`
- `timeUnixNano`enthält die CloudWatch `endTime`
- `sum` enthält die Summen-Statistik.
- `count`enthält die SampleCount Statistik.
- `quantile_values` enthält zwei `valueAtQuantile.value`-Objekte:
  - `valueAtQuantile.quantile = 0.0` mit `valueAtQuantile.value = Min value`
  - `valueAtQuantile.quantile = 0.99` mit `valueAtQuantile.value = p99 value`
  - `valueAtQuantile.quantile = 0.999` mit `valueAtQuantile.value = p99.9 value`
  - `valueAtQuantile.quantile = 1.0` mit `valueAtQuantile.value = Max value`

Ressourcen, die den Metrik-Stream nutzen, können die Durchschnittsstatistik als Sum/ berechnen. SampleCount

Einheiten umrechnen

CloudWatch Einheiten werden der Variante des Unified Codes für Maßeinheiten zugeordnet, bei der Groß- und Kleinschreibung berücksichtigt wird, wie in der folgenden Tabelle dargestellt. Weitere Informationen finden Sie unter Der einheitliche Code für Maßeinheiten.

CloudWatch	OpenTelemetry
Sekunde	S
Sekunde oder Sekunden	S
Mikrosekunden	wir
Millisekunden	ms
Bytes	Von
Kilobytes	kBy
Megabyte	Mby
Gigabytes	GBy
Terabytes	TBy
Bits	Bit
Kilobits	kbit
Megabits	MBit
Gigabits	GBit
Terabits	Tbit
Prozent	%
Anzahl	{Count}

CloudWatch	OpenTelemetry
None	1

Einheiten, die mit einem Schrägstrich kombiniert werden, werden zugeordnet, indem die OpenTelemetry Konvertierung beider Einheiten angewendet wird. Beispielsweise wird Bytes/Sekunde auf By/s abgebildet.

Wie analysiert man 1.0.0-Nachrichten OpenTelemetry

Dieser Abschnitt enthält Informationen, die Ihnen den Einstieg in das OpenTelemetry Parsen von 1.0.0 erleichtern sollen.

Zunächst sollten Sie sich sprachspezifische Bindungen besorgen, mit denen Sie OpenTelemetry 1.0.0-Nachrichten in Ihrer bevorzugten Sprache analysieren können.

So erhalten Sie sprachspezifische Bindungen

- Die Schritte hängen von Ihrer bevorzugten Sprache ab.
  - Um Java zu verwenden, fügen Sie Ihrem Java-Projekt die folgende Maven-Abhängigkeit hinzu: Java >> 0.14.1. OpenTelemetry
  - Gehen Sie folgendermaßen vor, um eine andere Sprache zu verwenden:

    a. Stellen Sie sicher, dass Ihre Sprache unterstützt wird, indem Sie die Liste unter Generieren Ihrer Klassen überprüfen.

    b. Installieren Sie den Protobuf-Compiler, indem Sie die Schritte unter Protokollpuffer herunterladen befolgen.

    c. Laden Sie die OpenTelemetry ProtoBuf 0.7.0-Definitionen in der Release-Version 1.0.0 herunter.

    d. Vergewissern Sie sich, dass Sie sich im Stammordner der heruntergeladenen OpenTelemetry ProtoBuf 0.7.0-Definitionen befinden. Dann erstellen Sie einen `src`-Ordner und führen Sie den Befehl aus, um sprachspezifische Bindungen zu generieren. Weitere Informationen finden Sie unter Generieren Ihrer Klassen.

    Nachfolgend finden Sie ein Beispiel für das Generieren von Javascript-Bindungen.

```
protoc --proto_path=./ --js_out=import_style=commonjs,binary:src \
opentelemetry/proto/common/v1/common.proto \
```

```
opentelemetry/proto/resource/v1/resource.proto \
opentelemetry/proto/metrics/v1/metrics.proto \
opentelemetry/proto/collector/metrics/v1/metrics_service.proto
```

Der folgende Abschnitt enthält Beispiele für die Verwendung der sprachspezifischen Bindungen, die Sie mit den vorherigen Anweisungen erstellen können.

Java

```
package com.example;

import io.opentelemetry.proto.collector.metrics.v1.ExportMetricsServiceRequest;

import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

public class MyOpenTelemetryParser {

    public List<ExportMetricsServiceRequest> parse(InputStream inputStream) throws
 IOException {
        List<ExportMetricsServiceRequest> result = new ArrayList<>();

        ExportMetricsServiceRequest request;
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        while ((request =
 ExportMetricsServiceRequest.parseDelimitedFrom(inputStream)) != null) {
            // Do whatever we want with the parsed message
            result.add(request);
        }

        return result;
    }
}
```

JavaScript

In diesem Beispiel wird davon ausgegangen, dass der Stammordner mit den generierten Bindungen `./` ist.

Das Datenargument der Funktion `parseRecord` kann einer der folgenden Typen sein:

- `Uint8Array` dies ist optimal.
- `Buffer` optimal unter Knoten
- `Array.number` 8-Bit-Ganzzahlen

```
const pb = require('google-protobuf')
const pbMetrics =
    require('./opentelemetry/proto/collector/metrics/v1/metrics_service_pb')

function parseRecord(data) {
    const result = []

    // Loop until we've read all the data from the buffer
    while (data.length) {
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        const reader = new pb.BinaryReader(data)
        const messageLength = reader.decoder_.readUnsignedVarint32()
        const messageFrom = reader.decoder_.cursor_
        const messageTo = messageFrom + messageLength

        // Extract the current `ExportMetricsServiceRequest` message to parse
        const message = data.subarray(messageFrom, messageTo)

        // Parse the current message using the ProtoBuf library
        const parsed =
            pbMetrics.ExportMetricsServiceRequest.deserializeBinary(message)

        // Do whatever we want with the parsed message
        result.push(parsed.toObject())
```

```
        // Shrink the remaining buffer, removing the already parsed data
        data = data.subarray(messageTo)
    }

    return result
}
```

Python

Sie müssen die `var-int`-Trennzeichen selbst lesen oder die internen Methoden `_VarintBytes(size)` und `_DecodeVarint32(buffer, position)` verwenden. Diese geben die Position im Puffer direkt nach den Größenbytes zurück. Die Leseseite erstellt einen neuen Puffer, der darauf beschränkt ist, nur die Bytes der Nachricht zu lesen.

```
size = my_metric.ByteSize()
f.write(_VarintBytes(size))
f.write(my_metric.SerializeToString())
msg_len, new_pos = _DecodeVarint32(buf, 0)
msg_buf = buf[new_pos:new_pos+msg_len]
request = metrics_service_pb.ExportMetricsServiceRequest()
request.ParseFromString(msg_buf)
```

Go

Verwenden Sie `Buffer.DecodeMessage()`.

C#

Verwenden Sie `CodedInputStream`. Diese Klasse kann Nachrichten mit größenbegrenztem Abstand lesen.

C++

Die in `google/protobuf/util/delimited_message_util.h` beschriebenen Funktionen können größenbegrenzte Nachrichten lesen.

Andere Sprachen

Informationen zu anderen Sprachen finden Sie unter Herunterladen von Protokollpuffern.

Bedenken Sie bei der Implementierung des Parsers, dass ein Kinesis-Datensatz mehrere `ExportMetricsServiceRequest` Protokollpuffer-Nachrichten enthalten kann, von denen jede mit einem Header mit einem `UnsignedVarInt32` beginnt, das die Datensatzlänge in Byte angibt.

OpenTelemetry 0.7.0-Format

OpenTelemetry ist eine Sammlung von Tools, APIs und SDKs. Sie können damit Telemetriedaten (Metriken, Protokolle und Traces) für Analysen instrumentieren, generieren, sammeln und exportieren. OpenTelemetry ist Teil der Cloud Native Computing Foundation. Weitere Informationen finden Sie unter OpenTelemetry.

Informationen zur vollständigen OpenTelemetry 0.7.0-Spezifikation finden Sie in der Version v0.7.0.

Ein Kinesis-Datensatz kann eine oder mehrere `ExportMetricsServiceRequest` OpenTelemetry Datenstrukturen enthalten. Jede Datenstruktur beginnt mit einem Header mit einem `UnsignedVarInt32`, das die Datensatzlänge in Byte angibt. Jede `ExportMetricsServiceRequest` kann Daten von mehreren Metriken gleichzeitig enthalten.

Das Folgende ist eine Zeichenkettendarstellung der Botschaft der `ExportMetricsServiceRequest` OpenTelemetry Datenstruktur. OpenTelemetry serialisiert das Binärprotokoll von Google Protocol Buffers, und dieses ist nicht für Menschen lesbar.

```
resource_metrics {
  resource {
    attributes {
      key: "cloud.provider"
      value {
        string_value: "aws"
      }
    }
    attributes {
      key: "cloud.account.id"
      value {
        string_value: "2345678901"
      }
    }
    attributes {
      key: "cloud.region"
      value {
        string_value: "us-east-1"
      }
    }
    attributes {
      key: "aws.exporter.arn"
      value {
```

```
        string_value: "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/
MyMetricStream"
      }
    }
  }
  instrumentation_library_metrics {
    metrics {
      name: "amazonaws.com/AWS/DynamoDB/ConsumedReadCapacityUnits"
      unit: "1"
      double_summary {
        data_points {
          labels {
            key: "Namespace"
            value: "AWS/DynamoDB"
          }
          labels {
            key: "MetricName"
            value: "ConsumedReadCapacityUnits"
          }
          labels {
            key: "TableName"
            value: "MyTable"
          }
          start_time_unix_nano: 1604948400000000000
          time_unix_nano: 1604948460000000000
          count: 1
          sum: 1.0
          quantile_values {
            quantile: 0.0
            value: 1.0
          }
          quantile_values {
            quantile: 0.95
            value: 1.0
          }
          quantile_values {
            quantile: 0.99
            value: 1.0
          }
          quantile_values {
            quantile: 1.0
            value: 1.0
          }
        }
```

```
        data_points {
          labels {
            key: "Namespace"
            value: "AWS/DynamoDB"
          }
          labels {
            key: "MetricName"
            value: "ConsumedReadCapacityUnits"
          }
          labels {
            key: "TableName"
            value: "MyTable"
          }
          start_time_unix_nano: 1604948460000000000
          time_unix_nano: 1604948520000000000
          count: 2
          sum: 5.0
          quantile_values {
            quantile: 0.0
            value: 2.0
          }
          quantile_values {
            quantile: 1.0
            value: 3.0
          }
        }
      }
    }
  }
}
```

Objekt der obersten Ebene zur Serialisierung von Metrikdaten OpenTelemetry

`ExportMetricsServiceRequest`ist der Wrapper der obersten Ebene zum Serialisieren einer Export-Payload. OpenTelemetry Es enthält eine oder mehrere `ResourceMetrics`.

```
message ExportMetricsServiceRequest {
  // An array of ResourceMetrics.
  // For data coming from a single resource this array will typically contain one
  // element. Intermediary nodes (such as OpenTelemetry Collector) that receive
  // data from multiple origins typically batch the data before forwarding further and
  // in that case this array will contain multiple elements.
  repeated opentelemetry.proto.metrics.v1.ResourceMetrics resource_metrics = 1;
```

```
}
```

`ResourceMetrics`ist das Objekt der obersten Ebene zur Darstellung von Objekten. MetricData

```
// A collection of InstrumentationLibraryMetrics from a Resource.
message ResourceMetrics {
  // The resource for the metrics in this message.
  // If this field is not set then no resource info is known.
  opentelemetry.proto.resource.v1.Resource resource = 1;

  // A list of metrics that originate from a resource.
  repeated InstrumentationLibraryMetrics instrumentation_library_metrics = 2;
}
```

Ressourcenobjekt

Ein `Resource`-Objekt ist ein Wertpaarobjekt, das Informationen über die Ressource enthält, die die Metriken generiert hat. Bei Metriken, die von AWS erstellt wurden, enthält die Datenstruktur den Amazon-Ressourcennamen (ARN) der Ressource, die sich auf die Metrik bezieht, z. B. eine EC2-Instance oder ein S3 Bucket.

Das `Resource`-Objekt enthält ein Attribut namens `attributes`, das eine Liste von Schlüssel-Wert-Paaren speichert.

- `cloud.account.id` enthält die Konto-ID
- `cloud.region` enthält die Region
- `aws.exporter.arn` enthält den Metrik-Stream-ARN
- `cloud.provider` ist immer `aws`.

```
// Resource information.
message Resource {
  // Set of labels that describe the resource.
  repeated opentelemetry.proto.common.v1.KeyValue attributes = 1;

  // dropped_attributes_count is the number of dropped attributes. If the value is 0,
  // no attributes were dropped.
  uint32 dropped_attributes_count = 2;
}
```

Das Objekt InstrumentationLibraryMetrics

Das Feld instrumentation_library wird nicht ausgefüllt. Wir füllen nur das Metrikfeld aus, das wir exportieren.

```
// A collection of Metrics produced by an InstrumentationLibrary.
message InstrumentationLibraryMetrics {
  // The instrumentation library information for the metrics in this message.
  // If this field is not set then no library info is known.
  opentelemetry.proto.common.v1.InstrumentationLibrary instrumentation_library = 1;
  // A list of metrics that originate from an instrumentation library.
  repeated Metric metrics = 2;
}
```

Metrikobjekt

Das Metrikobjekt enthält ein `DoubleSummary`-Datenfeld, das eine Liste von `DoubleSummaryDataPoint` enthält.

```
message Metric {
  // name of the metric, including its DNS name prefix. It must be unique.
  string name = 1;

  // description of the metric, which can be used in documentation.
  string description = 2;

  // unit in which the metric value is reported. Follows the format
  // described by http://unitsofmeasure.org/ucum.html.
  string unit = 3;

  oneof data {
    IntGauge int_gauge = 4;
    DoubleGauge double_gauge = 5;
    IntSum int_sum = 6;
    DoubleSum double_sum = 7;
    IntHistogram int_histogram = 8;
    DoubleHistogram double_histogram = 9;
    DoubleSummary double_summary = 11;
  }
}

message DoubleSummary {
  repeated DoubleSummaryDataPoint data_points = 1;
```

```
}
```

Das MetricDescriptor Objekt

Das MetricDescriptor Objekt enthält Metadaten. Weitere Informationen finden Sie unter metrics.proto on. GitHub

Für metrische Streams MetricDescriptor hat der den folgenden Inhalt:

- `name` wird `amazonaws.com/`*`metric_namespace`*`/`*`metric_name`* sein.
- `description` wird leer sein.
- `unit` wird gefüllt, indem die Einheit des metrischen Datums auf die Variante des einheitlichen Codes für Maßeinheiten, bei der die Groß-/Kleinschreibung beachtet wird, abgebildet wird. Weitere Informationen finden Sie unter Übersetzungen im Format OpenTelemetry 0.7.0 und Der einheitliche Code für Maßeinheiten.
- `type` wird `SUMMARY` sein.

Das DoubleSummaryDataPoint Objekt

Das DoubleSummaryDataPoint Objekt enthält den Wert eines einzelnen Datenpunkts in einer Zeitreihe in einer DoubleSummary Metrik.

```
// DoubleSummaryDataPoint is a single data point in a timeseries that describes the
// time-varying values of a Summary metric.
message DoubleSummaryDataPoint {
  // The set of labels that uniquely identify this timeseries.
  repeated opentelemetry.proto.common.v1.StringKeyValue labels = 1;

  // start_time_unix_nano is the last time when the aggregation value was reset
  // to "zero". For some metric types this is ignored, see data types for more
  // details.
  //
  // The aggregation value is over the time interval (start_time_unix_nano,
  // time_unix_nano].
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  //
  // Value of 0 indicates that the timestamp is unspecified. In that case the
  // timestamp may be decided by the backend.
```

```
  fixed64 start_time_unix_nano = 2;

  // time_unix_nano is the moment when this aggregation value was reported.
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  fixed64 time_unix_nano = 3;

  // count is the number of values in the population. Must be non-negative.
  fixed64 count = 4;

  // sum of the values in the population. If count is zero then this field
  // must be zero.
  double sum = 5;

  // Represents the value at a given quantile of a distribution.
  //
  // To record Min and Max values following conventions are used:
  // - The 1.0 quantile is equivalent to the maximum value observed.
  // - The 0.0 quantile is equivalent to the minimum value observed.
  message ValueAtQuantile {
    // The quantile of a distribution. Must be in the interval
    // [0.0, 1.0].
    double quantile = 1;

    // The value at the given quantile of a distribution.
    double value = 2;
  }

  // (Optional) list of values at different quantiles of the distribution calculated
  // from the current snapshot. The quantiles must be strictly increasing.
  repeated ValueAtQuantile quantile_values = 6;
}
```

Weitere Informationen finden Sie unter Übersetzungen im Format OpenTelemetry 0.7.0.

Übersetzungen im Format OpenTelemetry 0.7.0

CloudWatch führt einige Transformationen durch, um CloudWatch Daten in OpenTelemetry ein Format zu bringen.

Namespace, Metriknamen und Dimensionen übersetzen

Diese Attribute sind Schlüssel-Wert-Paare, die im Mapping kodiert sind.

- Ein Paar enthält den Namespace der Metrik
- Ein Paar enthält den Namen der Metrik
- CloudWatch Speichert für jede Dimension das folgende Paar: `metricDatum.Dimensions[i].Name, metricDatum.Dimensions[i].Value`

Durchschnitt, Summe SampleCount, Min und Max werden übersetzt

Der Zusammenfassungsdatenpunkt ermöglicht CloudWatch den Export all dieser Statistiken unter Verwendung eines Datenpunkts.

- `startTimeUnixNano`enthält die CloudWatch `startTime`
- `timeUnixNano`enthält die CloudWatch `endTime`
- `sum` enthält die Summen-Statistik.
- `count`enthält die SampleCount Statistik.
- `quantile_values` enthält zwei `valueAtQuantile.value`-Objekte:
  - `valueAtQuantile.quantile = 0.0` mit `valueAtQuantile.value = Min value`
  - `valueAtQuantile.quantile = 0.99` mit `valueAtQuantile.value = p99 value`
  - `valueAtQuantile.quantile = 0.999` mit `valueAtQuantile.value = p99.9 value`
  - `valueAtQuantile.quantile = 1.0` mit `valueAtQuantile.value = Max value`

Ressourcen, die den Metrik-Stream nutzen, können die Durchschnittsstatistik als Sum/ berechnen. SampleCount

Einheiten umrechnen

CloudWatch Einheiten werden der Variante des Unified Codes für Maßeinheiten zugeordnet, bei der Groß- und Kleinschreibung berücksichtigt wird, wie in der folgenden Tabelle dargestellt. Weitere Informationen finden Sie unter Der einheitliche Code für Maßeinheiten.

CloudWatch	OpenTelemetry
Sekunde	S
Sekunde oder Sekunden	S
Mikrosekunde	wir

CloudWatch	OpenTelemetry
Millisekunden	ms
Bytes	Von
Kilobytes	kBy
Megabyte	Mby
Gigabytes	GBy
Terabytes	TBy
Bits	Bit
Kilobits	kbit
Megabits	MBit
Gigabits	GBit
Terabits	Tbit
Prozent	%
Anzahl	{Count}
None	1

Einheiten, die mit einem Schrägstrich kombiniert werden, werden zugeordnet, indem die OpenTelemetry Konvertierung beider Einheiten angewendet wird. Beispielsweise wird Bytes/Sekunde auf By/s abgebildet.

Wie analysiert man 0.7.0-Nachrichten OpenTelemetry

Dieser Abschnitt enthält Informationen, die Ihnen den Einstieg in das OpenTelemetry Parsen von 0.7.0 erleichtern sollen.

Zunächst sollten Sie sich sprachspezifische Bindungen besorgen, mit denen Sie OpenTelemetry 0.7.0-Nachrichten in Ihrer bevorzugten Sprache analysieren können.

So erhalten Sie sprachspezifische Bindungen

- Die Schritte hängen von Ihrer bevorzugten Sprache ab.
  - Um Java zu verwenden, fügen Sie Ihrem Java-Projekt die folgende Maven-Abhängigkeit hinzu: Java >> 0.14.1. OpenTelemetry
  - Gehen Sie folgendermaßen vor, um eine andere Sprache zu verwenden:

    a. Stellen Sie sicher, dass Ihre Sprache unterstützt wird, indem Sie die Liste unter Generieren Ihrer Klassen überprüfen.

    b. Installieren Sie den Protobuf-Compiler, indem Sie die Schritte unter Protokollpuffer herunterladen befolgen.

    c. Laden Sie die OpenTelemetry ProtoBuf 0.7.0-Definitionen in Version v0.7.0 herunter.

    d. Vergewissern Sie sich, dass Sie sich im Stammordner der heruntergeladenen OpenTelemetry 0.7.0-Definitionen befinden. ProtoBuf Dann erstellen Sie einen `src`-Ordner und führen Sie den Befehl aus, um sprachspezifische Bindungen zu generieren. Weitere Informationen finden Sie unter Generieren Ihrer Klassen.

    Nachfolgend finden Sie ein Beispiel für das Generieren von Javascript-Bindungen.

```
protoc --proto_path=./ --js_out=import_style=commonjs,binary:src \
opentelemetry/proto/common/v1/common.proto \
opentelemetry/proto/resource/v1/resource.proto \
opentelemetry/proto/metrics/v1/metrics.proto \
opentelemetry/proto/collector/metrics/v1/metrics_service.proto
```

Der folgende Abschnitt enthält Beispiele für die Verwendung der sprachspezifischen Bindungen, die Sie mit den vorherigen Anweisungen erstellen können.

Java

```
package com.example;

import io.opentelemetry.proto.collector.metrics.v1.ExportMetricsServiceRequest;

import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;
```

```
public class MyOpenTelemetryParser {

    public List<ExportMetricsServiceRequest> parse(InputStream inputStream) throws
 IOException {
        List<ExportMetricsServiceRequest> result = new ArrayList<>();

        ExportMetricsServiceRequest request;
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        while ((request =
 ExportMetricsServiceRequest.parseDelimitedFrom(inputStream)) != null) {
            // Do whatever we want with the parsed message
            result.add(request);
        }

        return result;
    }
}
```

JavaScript

In diesem Beispiel wird davon ausgegangen, dass der Stammordner mit den generierten Bindungen `./` ist.

Das Datenargument der Funktion `parseRecord` kann einer der folgenden Typen sein:

- `Uint8Array` dies ist optimal.
- `Buffer` optimal unter Knoten
- `Array.number` 8-Bit-Ganzzahlen

```
const pb = require('google-protobuf')
const pbMetrics =
    require('./opentelemetry/proto/collector/metrics/v1/metrics_service_pb')

function parseRecord(data) {
    const result = []
```

```
    // Loop until we've read all the data from the buffer
    while (data.length) {
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        const reader = new pb.BinaryReader(data)
        const messageLength = reader.decoder_.readUnsignedVarint32()
        const messageFrom = reader.decoder_.cursor_
        const messageTo = messageFrom + messageLength

        // Extract the current `ExportMetricsServiceRequest` message to parse
        const message = data.subarray(messageFrom, messageTo)

        // Parse the current message using the ProtoBuf library
        const parsed =
            pbMetrics.ExportMetricsServiceRequest.deserializeBinary(message)

        // Do whatever we want with the parsed message
        result.push(parsed.toObject())

        // Shrink the remaining buffer, removing the already parsed data
        data = data.subarray(messageTo)
    }

    return result
}
```

Python

Sie müssen die `var-int`-Trennzeichen selbst lesen oder die internen Methoden `_VarintBytes(size)` und `_DecodeVarint32(buffer, position)` verwenden. Diese geben die Position im Puffer direkt nach den Größenbytes zurück. Die Leseseite erstellt einen neuen Puffer, der darauf beschränkt ist, nur die Bytes der Nachricht zu lesen.

```
size = my_metric.ByteSize()
f.write(_VarintBytes(size))
f.write(my_metric.SerializeToString())
msg_len, new_pos = _DecodeVarint32(buf, 0)
```

```
msg_buf = buf[new_pos:new_pos+msg_len]
request = metrics_service_pb.ExportMetricsServiceRequest()
request.ParseFromString(msg_buf)
```

Go

Verwenden Sie `Buffer.DecodeMessage()`.

C#

Verwenden Sie `CodedInputStream`. Diese Klasse kann Nachrichten mit größenbegrenztem Abstand lesen.

C++

Die in `google/protobuf/util/delimited_message_util.h` beschriebenen Funktionen können größenbegrenzte Nachrichten lesen.

Andere Sprachen

Informationen zu anderen Sprachen finden Sie unter Herunterladen von Protokollpuffern.

Bedenken Sie bei der Implementierung des Parsers, dass ein Kinesis-Datensatz mehrere `ExportMetricsServiceRequest` Protokollpuffer-Nachrichten enthalten kann, von denen jede mit einem Header mit einem `UnsignedVarInt32` beginnt, das die Datensatzlänge in Byte angibt.

Fehlerbehebung

Wenn Sie keine Metrik-Daten am endgültigen Ziel sehen, überprüfen Sie Folgendes:

- Überprüfen Sie, ob der Metrik-Stream ausgeführt wird. Anweisungen zur Verwendung der CloudWatch Konsole zu diesem Zweck finden Sie unterBetrieb und Wartung von Metrik-Streams.
- Metriken, die vor mehr als zwei Tagen veröffentlicht wurden, werden nicht gestreamt. Um festzustellen, ob eine bestimmte Metrik gestreamt wird, stellen Sie die Metrik in der CloudWatch Konsole grafisch dar und überprüfen Sie, wie alt der letzte sichtbare Datenpunkt ist. Wenn es mehr als zwei Tage in der Vergangenheit liegt, wird es nicht von Metrik-Streams erfasst.
- Überprüfen Sie die Metriken, die vom Metrik-Stream ausgegeben werden. Suchen Sie in der CloudWatch Konsole unter Metrics im AWS/CloudWatch/MetricStreams-Namespace für die Metriken MetricUpdateTotalMetricUpdate, und. PublishErrorRate

- Wenn die PublishErrorRateMetrik hoch ist, stellen Sie sicher, dass das Ziel, das vom Firehose-Lieferstream verwendet wird, existiert und dass die in der Konfiguration des Metrik-Streams angegebene IAM-Rolle dem `CloudWatch` Dienstprinzipalberechtigungen zum Schreiben darauf gewährt. Weitere Informationen finden Sie unter Vertrauen zwischen CloudWatch und Firehose.
- Stellen Sie sicher, dass der Firehose-Lieferstream die Berechtigung hat, in das endgültige Ziel zu schreiben.
- Sehen Sie sich in Firehose Firehose-Konsole den Firehose-Lieferstream an, der für den Metrik-Stream verwendet wird, und überprüfen Sie auf der Registerkarte Überwachung, ob der Firehose-Lieferstream Daten empfängt.
- Vergewissern Sie sich, dass Sie Ihren Firehose-Lieferstream mit den richtigen Details konfiguriert haben.
- Überprüfen Sie alle verfügbaren Protokolle oder Metriken für das endgültige Ziel, in das der Firehose-Lieferstream schreibt.
- Um detailliertere Informationen zu erhalten, aktivieren Sie die CloudWatch Logs-Fehlerprotokollierung im Firehose-Lieferstream. Weitere Informationen finden Sie unter Amazon Data Firehose mithilfe von CloudWatch Protokollen überwachen.

Anzeigen der verfügbaren Metriken

Metriken werden zuerst nach dem Namespace und dann nach verschiedenen Dimensionskombinationen in jedem Namespace gruppiert. Beispiel: Sie können alle EC2-Metriken, EC2-Metriken zusammengefasst nach Instance oder EC2-Metriken zusammengefasst nach Auto-Scaling-Gruppe anzeigen.

Nur die AWS Dienste, die Sie verwenden, senden Messwerte an Amazon CloudWatch.

Eine Liste der AWS Dienste, an die Messwerte gesendet werden CloudWatch, finden Sie unterAWS Dienste, die CloudWatch Metriken veröffentlichen. Auf dieser Seite können Sie auch die Metriken und Dimensionen anzeigen, die von jedem dieser Services veröffentlicht werden.

> **Note**
>
> Metriken, für die in den letzten zwei Wochen keine neuen Datenpunkte vorlagen, werden nicht in der Konsole angezeigt. Sie werden auch nicht angezeigt, wenn Sie den Metriknamen oder die Dimensionsnamen in der Konsole in das Suchfeld auf der Registerkarte All metrics (Alle Metriken) eingeben, und sie werden nicht in den Ergebnissen eines Befehls vom Typ

list-metrics zurückgegeben. Diese Metriken lassen sich am besten mit den get-metric-statisticsBefehlen get-metric-dataoder in der abrufen AWS CLI.
Wenn die alte Metrik, die Sie anzeigen möchten, über eine aktuelle Metrik mit ähnlichen Dimensionen verfügt, können Sie diese aktuelle ähnliche Metrik anzeigen und dann die Registerkarte Source (Quelle) auswählen und die Felder für Metrikname und Dimension in die gewünschten Werte ändern. Außerdem kann der Zeitbereich in einen Zeitpunkt geändert werden, zu dem die Metrik gemeldet wurde.

Die folgenden Schritte helfen Ihnen beim Durchsuchen der Metrik-Namespaces, um Metriken zu finden und anzuzeigen. Sie können auch über gezielte Suchbegriffe nach Metriken suchen. Weitere Informationen finden Sie unter Nach verfügbaren Metriken suchen.

Wenn Sie in einem Konto surfen, das als Überwachungskonto mit CloudWatch kontenübergreifender Observability eingerichtet wurde, können Sie sich Metriken der Quellkonten ansehen, die mit diesem Überwachungskonto verknüpft sind. Wenn Metriken von Quellkonten angezeigt werden, wird auch die ID oder das Label des Kontos angezeigt, von dem sie stammen. Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.

So zeigen Sie verfügbare Metriken nach Namespaces und Dimension mithilfe der Konsole an:

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken) All metrics (Alle Metriken) aus.
3. Wählen Sie einen Metrik-Namespace aus (z. B. EC2 oder Lambda).
4. Wählen Sie eine Metrikdimension aus (z. B. Per-Instance Metrics (Metriken pro Instance) oder By Function Name (nach Funktionsname)).
5. Die Registerkarte Browse (Durchsuchen) zeigt alle Metriken für diese Dimension im Namespace an. Neben jedem Metriknamen befindet sich eine Informationsschaltfläche, über die Sie ein Popup mit der Definition der Metrik aufrufen können.

   Wenn es sich um ein Überwachungskonto mit CloudWatch kontenübergreifender Observability handelt, werden Ihnen auch die Metriken der Quellkonten angezeigt, die mit diesem Überwachungskonto verknüpft sind. In den Spalten Account label (Kontenbezeichnung) und Account id (Konto-ID) in der Tabelle wird angezeigt, aus welchem Konto die einzelnen Metriken stammen.

   Sie haben die folgenden Möglichkeiten:

a. Um die Tabelle sortieren, verwenden Sie die Spaltenüberschrift.
b. Um eine Metrik grafisch darzustellen, müssen Sie das Kontrollkästchen neben der Metrik aktivieren. Um alle Metriken auszuwählen, aktivieren Sie das Kontrollkästchen in der Kopfzeile der Tabelle.
c. Um nach Konto zu filtern, wählen Sie die Kontolabel oder die Konto-ID aus und wählen Sie dann Add to search (Zur Suche hinzufügen).
d. Um nach Ressource zu filtern, müssen Sie zunächst die Ressourcen-ID und dann die Option Zu Suche hinzufügen auswählen.
e. Um nach Metrik zu filtern, müssen Sie den Metriknamen und anschließend Add to search (Zur Suche hinzufügen) auswählen.

6. (Optional) Um dieses Diagramm zu einem CloudWatch Dashboard hinzuzufügen, wählen Sie Aktionen, Zum Dashboard hinzufügen aus.

Um verfügbare Metriken nach Kontonamespace, Dimension oder Metrik anzuzeigen, verwenden Sie AWS CLI

Verwenden Sie den Befehl list-metrics, um Metriken aufzulisten. CloudWatch Eine Liste der Namespaces, Metriken und Maße für alle Services, die Metriken veröffentlichen, finden Sie unter AWS Dienste, die CloudWatch Metriken veröffentlichen.

Der folgende Beispielbefehl listet alle Metriken für Amazon EC2 auf.

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

Es folgt eine Beispielausgabe.

```
{
  "Metrics" : [
    ...
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "NetworkOut"
```

```
    },
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "CPUUtilization"
    },
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "NetworkIn"
    },
    ...
  ]
}
```

So listen Sie alle verfügbaren Metriken für eine bestimmte Ressource auf

Im folgenden Beispiel werden der `AWS/EC2`-Namespace und die `InstanceId`-Dimension angegeben, um die Ergebnisse nur für die angegebene Instance anzuzeigen.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --dimensions
 Name=InstanceId,Value=i-1234567890abcdef0
```

So listen Sie eine Metrik für alle Ressourcen auf

Im folgenden Beispiel werden der `AWS/EC2`-Namespace und ein Metrikname angegeben, um die Ergebnisse nur für die angegebene Metrik anzuzeigen.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --metric-name CPUUtilization
```

Um Metriken von verknüpften Quellkonten in CloudWatch kontenübergreifender Observability abzurufen

Das folgende Beispiel wird in einem Überwachungskonto ausgeführt, um Metriken sowohl aus dem Überwachungskonto als auch aus allen verknüpften Quellkonten abzurufen. Wenn Sie `--include-linked-accounts` nicht hinzufügen, gibt der Befehl nur die Metriken des Überwachungskontos zurück.

```
aws cloudwatch list-metrics --include-linked-accounts
```

Um Metriken aus einem Quellkonto mit kontenübergreifender Observability abzurufen CloudWatch

Folgendes Beispiel wird in einem Überwachungskonto ausgeführt, um Metriken aus dem Quellkonto mit der ID 111122223333 abzurufen.

```
aws cloudwatch list-metrics --include-linked-accounts --owning-account "111122223333"
```

Nach verfügbaren Metriken suchen

Sie können innerhalb aller Metriken in Ihrem Konto mit gezielten Suchbegriffen suchen. Metriken, die in ihrem Namespace, dem Metriknamen oder den Dimensionen ein passendes Ergebnis haben, werden zurückgegeben.

Wenn es sich um ein Monitoring-Konto mit CloudWatch kontenübergreifender Observability handelt, suchen Sie auch nach Messwerten aus den Quellkonten, die mit diesem Monitoring-Konto verknüpft sind.

> **Note**
>
> Metriken, für die in den letzten zwei Wochen keine neuen Datenpunkte vorlagen, werden nicht in der Konsole angezeigt. Sie werden auch nicht angezeigt, wenn Sie den Metriknamen oder die Dimensionsnamen in der Konsole in das Suchfeld auf der Registerkarte All metrics (Alle Metriken) eingeben, und sie werden nicht in den Ergebnissen eines Befehls vom Typ list-metrics zurückgegeben. Diese Metriken lassen sich am besten mit den get-metric-statisticsBefehlen get-metric-dataoder in der abrufen. AWS CLI

Um nach verfügbaren Metriken zu suchen CloudWatch

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.

3. Geben Sie in das Suchfeld auf der Registerkarte All metrics (Alle Metriken) einen Suchbegriff, z. B. den Namen einer Metrik, den Namespace, einen Dimensionsnamen bzw. einen Dimensionswert oder einen Ressourcennamen ein. Damit werden alle Namespaces mit Metriken mit diesem Suchbegriff angezeigt.

   Wenn Sie beispielsweise nach **`volume`** suchen, werden die Namespaces mit Metriken mit diesem Begriff in ihren Namen angezeigt.

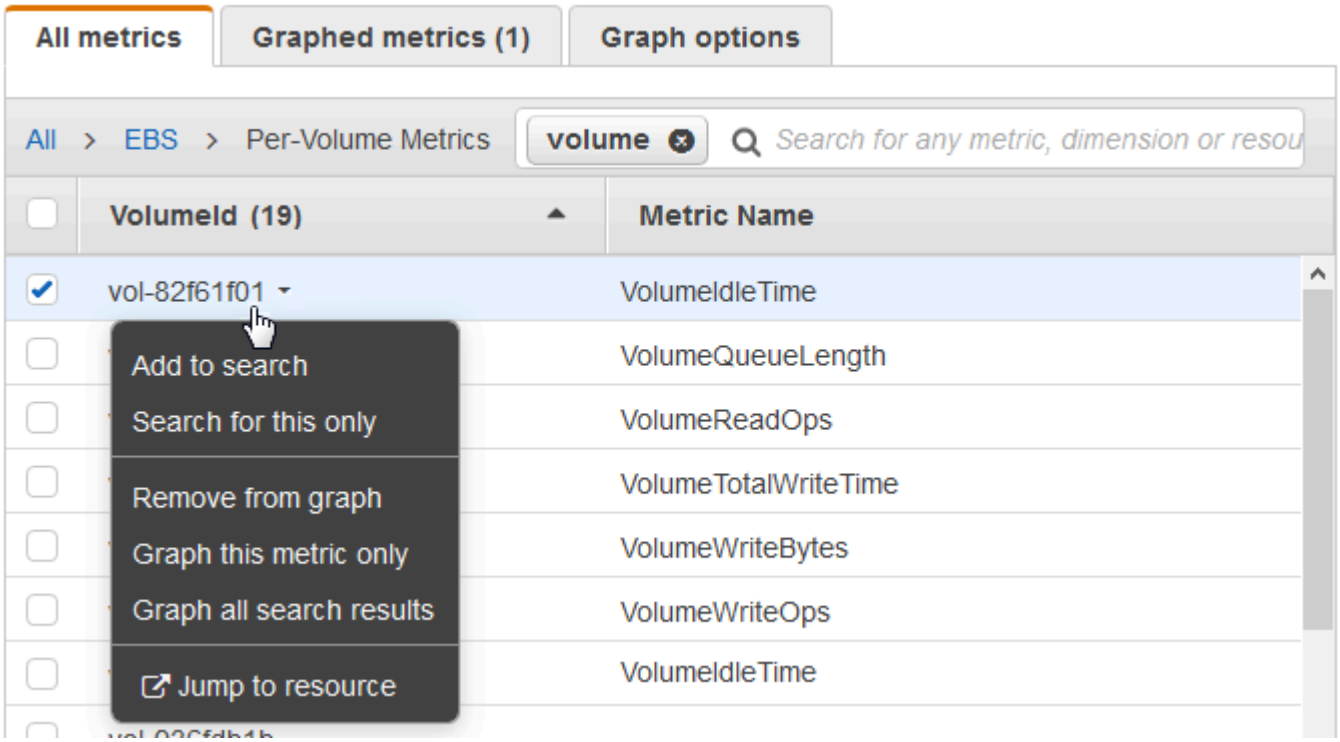
   Weitere Informationen zur Suche finden Sie unter Suchausdrücke in Diagrammen verwenden
4. Wählen Sie zur Diagrammdarstellung aller Suchergebnisse die Option Graph search (Graphensuche) aus.

   or

   Wählen Sie einen Namespace aus, um die Metriken aus diesem Namespace anzuzeigen. Sie können dann Folgendes durchführen:

   a. Um eine oder mehrere Metriken grafisch darzustellen, aktivieren Sie das Kontrollkästchen neben jeder Metrik. Um alle Metriken auszuwählen, aktivieren Sie das Kontrollkästchen in der Kopfzeile der Tabelle.
   b. Zeigen Sie zum Verfeinern der Suche mit dem Mauszeiger auf den Namen einer Metrik und wählen Sie Add to search (Zu Suche hinzufügen) oder Search for this only (Nur hiernach suchen) aus.
   c. Wenn Sie eine der Ressourcen in der entsprechenden Konsole anzeigen möchten, wählen Sie die Ressourcen-ID und anschließend Jump to resource (Zu Ressource springen) aus.
   d. Wenn Sie Hilfe für eine Metrik anzeigen möchten, wählen Sie den Metriknamen und anschließend What is this? (Was ist das?) aus.

   Die ausgewählten Metriken werden im Diagramm dargestellt.



5. (Optional) Verwenden Sie eine der Schaltflächen auf der Suchleiste, um den entsprechenden Teil des Suchbegriffs zu bearbeiten.

Grafisches Darstellen von Metriken

Verwenden Sie die CloudWatch Konsole, um von anderen AWS Diensten generierte Metrikdaten grafisch darzustellen. Hierdurch können Sie die Metrikaktivität für Ihre Services effizienter anzeigen. Die folgenden Verfahren beschreiben, wie Metriken grafisch dargestellt CloudWatch werden.

Inhalt

Grafisches Darstellen von Metriken

In der CloudWatch Konsole können Sie Metriken auswählen und Diagramme der Metrikdaten erstellen.

CloudWatch unterstützt die folgenden Statistiken zu Metriken: `Average`,`Minimum`,`Maximum`,`Sum`, und`SampleCount`. Weitere Informationen finden Sie unter Statistiken.

Sie können Ihre Daten in verschiedenen Detailebenen anzeigen. Sie können beispielsweise eine Ein-Minuten-Ansicht auswählen. Dies kann nützlich sein, wenn Sie Fehler beheben. Sie können auch eine weniger detaillierte Ein-Stunden-Ansicht auswählen. Dies kann nützlich sein, wenn Sie einen größeren Zeitraum anzeigen (z. B. 3 Tage), damit Sie Trends über die Zeit anzeigen können. Weitere Informationen finden Sie unter Zeiträume.

Wenn Sie ein Konto verwenden, das als Überwachungskonto für CloudWatch kontenübergreifende Observability eingerichtet ist, können Sie Metriken der Quellkonten, die mit diesem Überwachungskonto verknüpft sind, grafisch darstellen. Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.

Erstellen eines Diagramms

So stellen Sie eine Metrik grafisch dar

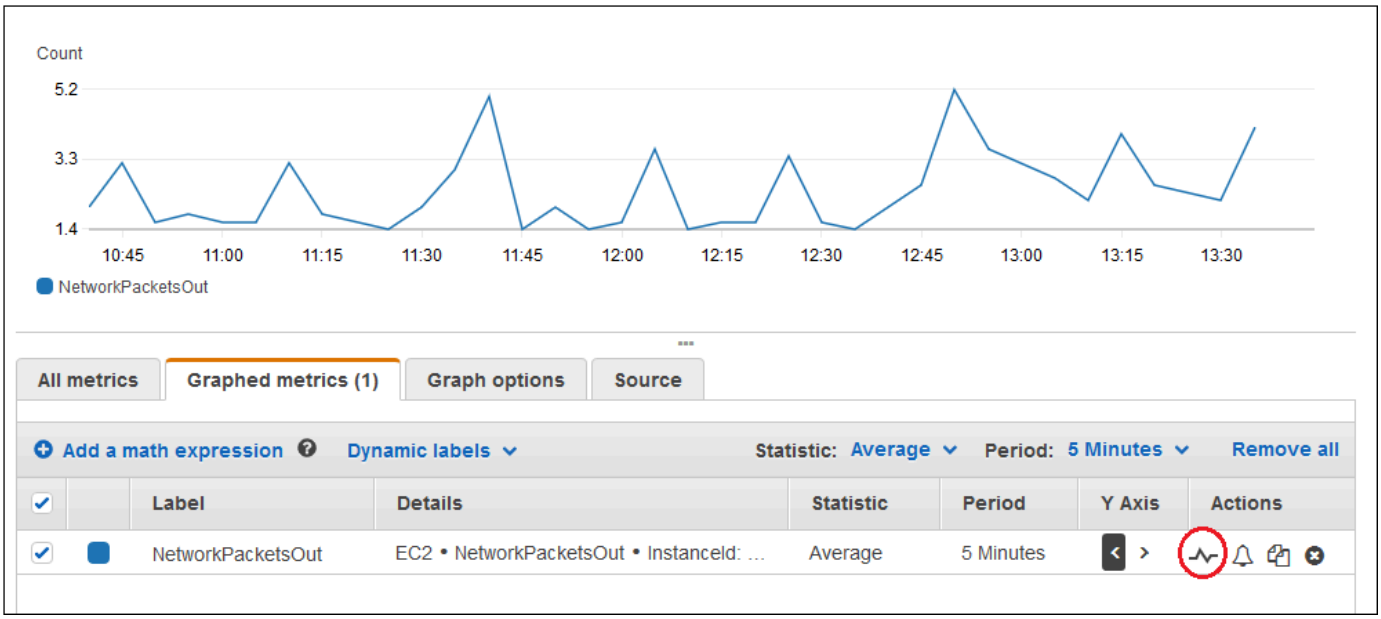
1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken) All metrics (Alle Metriken) aus.
3. Geben Sie in der Registerkarte Suche einen Suchbegriff in das Suchfeld ein, z. B. einen Metriknamen, die Konto-ID oder einen Ressourcennamen.

   Wenn Sie beispielsweise nach der Metrik `CPUUtilization` suchen, werden Namespaces und Dimensionen mit dieser Metrik angezeigt.
4. Wählen Sie eines der Suchergebnisse aus, um die Metriken anzuzeigen.
5. Um eine oder mehrere Metriken grafisch darzustellen, aktivieren Sie das Kontrollkästchen neben jeder Metrik. Um alle Metriken auszuwählen, aktivieren Sie das Kontrollkästchen in der Kopfzeile der Tabelle.
6. (Optional) Um den Diagrammtyp zu ändern, wählen Sie die Registerkarte Optionen. Sie können dann zwischen einem Liniendiagramm, einem gestapelten Flächendiagramm, einer Zahlenanzeige, einem Maßstab, einem Balkendiagramm oder einem Kreisdiagramm wählen.
7. Wählen Sie die Registerkarte Graphed metrics (Grafisch dargestellte Metriken) aus.

8. (Optional) Um die im Diagramm verwendete Statistik zu ändern, wählen Sie die neue Statistik in der Spalte Statistik neben dem Metriknamen aus.

   Weitere Informationen zu CloudWatch Statistiken finden Sie unterCloudWatch Definitionen von Statistiken. Weitere Informationen zu den pxx-Perzentilstatistiken finden Sie unter Perzentile.

9. (Optional) Um ein Anomalieerkennungsband hinzuzufügen, das die für die Metrik erwarteten Werte anzeigt, wählen Sie das Anomalieerkennungssymbol unterhalb von Actions (Aktionen) neben der Metrik aus.



   CloudWatch verwendet die jüngsten historischen Daten der Metrik für bis zu zwei Wochen, um ein Modell für erwartete Werte zu berechnen. Anschließend wird dieser Bereich erwarteter Werte als Band in der Grafik angezeigt. CloudWatch fügt unter der Metrik eine neue Zeile mit der Bezeichnung ANOMALY_DETECTION_BAND hinzu, um den mathematischen Ausdruck für das Band zur Erkennung von Anomalien anzuzeigen. Wenn aktuelle historische Daten vorhanden sind, wird Ihnen sofort eine Vorversion des Anomalieerkennungsbands angezeigt. Dabei handelt es sich um eine Approximation des Anomalieerkennungsbands, das vom Modell generiert wird. Es dauert bis zu 15 Minuten, bis das tatsächliche Anomalieerkennungsband angezeigt wird.

   CloudWatch Erstellt standardmäßig die Ober- und Untergrenzen der Bandbreite der erwarteten Werte mit dem Standardwert 2 für den Bandschwellenwert. Wenn Sie diese Anzahl ändern möchten, bearbeiten Sie den Wert am Ende der Formel unter Details für das Band.

   - (Optional) Sie können auch Edit model (Modell bearbeiten) auswählen, um die Art der Berechnung des Anomalieerkennungsmodells zu ändern. Sie können frühere und

zukünftige Zeiträume von der Verwendung beim Training für die Berechnung des Modells ausschließen. Es ist wichtig, dass Sie ungewöhnliche Ereignisse wie Systemausfälle, Bereitstellungen und Feiertage aus den Trainingsdaten ausschließen. Sie können auch die Zeitzone angeben, die das Modell verwenden soll, um Zeitumstellungen (Sommer- und Winterzeit) zu berücksichtigen.

Weitere Informationen finden Sie unter Verwendung der CloudWatch Anomalieerkennung.

Um das Modell aus dem Diagramm auszublenden, entfernen Sie das Häkchen aus der Zeile mit der `ANOMALY_DETECTION_BAND`-Funktion oder wählen das Symbol `X` aus. Um das Modell vollständig zu löschen, wählen Sie Edit model (Modell bearbeiten), Delete model (Modell löschen) aus.

10. (Optional) Während Sie die Metriken für das Diagramm auswählen, können Sie für jede Metrik eine dynamische Beschriftung angeben, die in der Diagrammlegende angezeigt wird. Dynamische Beschriftungen zeigen eine Statistik über die Metrik an und werden automatisch aktualisiert, wenn das Dashboard oder das Diagramm aktualisiert wird. Um ein dynamisches Label hinzuzufügen, wählen Sie grafisch dargestellte Metriken, Dynamisches Label hinzufügen.

    Standardmäßig werden die dynamischen Werte, die Sie der Beschriftung hinzufügen, am Anfang der Beschriftung angezeigt. Sie können dann den Wert Label (Beschriftung) für die Metrik wählen, um die Beschriftung zu bearbeiten. Weitere Informationen finden Sie unter Dynamische Labels verwenden.

11. Wenn Sie weitere Informationen zu der Metrik anzeigen möchten, die in Form eines Diagramm angezeigt wird, zeigen Sie mit der Maus auf die Legende.

12. Horizontale Anmerkungen können den Benutzern eines Diagramms helfen, effizienter zu erkennen, ob eine Metrik Spitzenwerte auf einer bestimmten Stufe erreicht hat oder ob eine Metrik innerhalb des vordefinierten Bereichs liegt. Um eine horizontale Anmerkung hinzuzufügen, wählen Sie die Registerkarte Optionen und dann Horizontale Anmerkung hinzufügen:

    a. Geben Sie unter Label (Bezeichnung) eine Bezeichnung für die Anmerkung ein.

    b. Geben Sie unter Value (Wert) den Metrikwert ein, an dem die horizontale Anmerkung angezeigt wird.

    c. Geben Sie unter Fill an, ob die Füllschattierung bei der Anmerkung verwendet werden soll. Wählen Sie beispielsweise `Above` oder `Below` als entsprechend zu füllenden Bereich aus. Wenn Sie `Between` angeben, wird ein anderes `Value`-Feld angezeigt und der Bereich des Diagramms zwischen zwei Werten wird gefüllt.

d. Geben Sie als Axis an, ob sich die Nummern in `Value` auf die der linken Y-Achse oder der rechten Y-Achse zugewiesenen Metrik beziehen, wenn das Diagramm mehrere Metriken enthält.

   Sie können die Füllfarbe einer Anmerkung ändern, indem Sie das Farbquadrat in der linken Spalte neben der Anmerkung auswählen.

Wiederholen Sie die Schritte, um einem Diagramm mehrere horizontale Anmerkungen hinzuzufügen.

Um eine Anmerkung auszublenden, deaktivieren Sie das Kontrollkästchen in der linken Spalte für diese Anmerkung.

Um eine Anmerkung zu löschen, wählen Sie das x in der Spalte Actions aus.

13. Wenn Sie eine URL für Ihr Diagramm abrufen möchten, wählen Sie Actions und Share aus. Kopieren Sie die URL zum Speichern oder Freigeben.
14. Um Ihr Diagramm einem Dashboard hinzuzufügen, wählen Sie Actions und Add to dashboard aus.

Erstellen eines Diagramms mit Metriken aus einer anderen Datenquelle

Sie können ein Diagramm erstellen, das Ressourcen aus anderen Datenquellen als CloudWatch anzeigt. Weitere Informationen zum Erstellen von Verbindungen zu diesen anderen Datenquellen finden Sie unter Metriken aus anderen Datenquellen abfragen.

So erstellen Sie ein Diagramms mit Metriken aus einer anderen Datenquelle

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken) All metrics (Alle Metriken) aus.
3. Wählen Sie die Registerkarte Abfrage mit mehreren Quellen.
4. Wählen Sie für Datenquelle die Datenquelle, die Sie verwenden möchten.

   Wenn Sie noch keine Verbindung zu der gewünschten Datenquelle hergestellt haben, wählen Sie Datenquellen erstellen und verwalten und dann Datenquellen erstellen und verwalten. Informationen zum weiteren Ablauf dieses Prozesses zur Erstellung von Datenquellen finden Sie unter Mit einem Assistenten eine Verbindung zu einer vordefinierten Datenquelle herstellen.

5. Der Assistent oder der Abfragen-Editor fordert Sie auf, die für die Abfrage erforderlichen Informationen einzugeben. Der Workflow ist für jede Datenquelle unterschiedlich und auf die Datenquelle zugeschnitten. Für Amazon Managed Service für Prometheus und Prometheus-Datenquellen wird beispielsweise ein PromQL-Abfrage-Editor-Feld mit einem Abfrage-Assistenten angezeigt.
6. Wenn Sie mit der Erstellung der Abfrage fertig sind, wählen Sie Diagramm-Abfrage.

   Das Diagramm wird mit Metriken aus der Abfrage gefüllt.
7. (Optional) Horizontale Anmerkungen können den Benutzern eines Diagramms helfen, effizienter zu erkennen, ob eine Metrik Spitzenwerte auf einer bestimmten Stufe erreicht hat oder ob eine Metrik innerhalb des vordefinierten Bereichs liegt. Um eine horizontale Anmerkung hinzuzufügen, wählen Sie die Registerkarte Optionen und dann Horizontale Anmerkung hinzufügen:

   a. Geben Sie unter Label (Bezeichnung) eine Bezeichnung für die Anmerkung ein.
   b. Geben Sie unter Value (Wert) den Metrikwert ein, an dem die horizontale Anmerkung angezeigt wird.
   c. Geben Sie unter Fill an, ob die Füllschattierung bei der Anmerkung verwendet werden soll. Wählen Sie beispielsweise `Above` oder `Below` als entsprechend zu füllenden Bereich aus. Wenn Sie `Between` angeben, wird ein anderes `Value`-Feld angezeigt und der Bereich des Diagramms zwischen zwei Werten wird gefüllt.
   d. Geben Sie als Axis an, ob sich die Nummern in `Value` auf die der linken Y-Achse oder der rechten Y-Achse zugewiesenen Metrik beziehen, wenn das Diagramm mehrere Metriken enthält.

      Sie können die Füllfarbe einer Anmerkung ändern, indem Sie das Farbquadrat in der linken Spalte neben der Anmerkung auswählen.

   Wiederholen Sie die Schritte, um einem Diagramm mehrere horizontale Anmerkungen hinzuzufügen.

   Um eine Anmerkung auszublenden, deaktivieren Sie das Kontrollkästchen in der linken Spalte für diese Anmerkung.

   Um eine Anmerkung zu löschen, wählen Sie das x in der Spalte Actions aus.
8. (Optional) Um dieses Diagramm zu einem Dashboard hinzuzufügen, wählen Sie Aktionen, Zu Dashboard hinzufügen.

Aktualisieren eines Diagramms

So aktualisieren Sie das Diagramm

1. Wenn Sie den Namen des Diagramms ändern möchten, wählen Sie das Bleistiftsymbol.
2. Wenn Sie den Zeitraum ändern möchten, müssen Sie einen der vordefinierten Werte oder custom (benutzerdefiniert) auswählen. Weitere Informationen finden Sie unter Den Zeitraum oder das Zeitzonenformat eines Diagramms ändern.
3. Wenn Sie die Statistik ändern möchten, wählen Sie die Registerkarte Graphed metrics aus. Wählen Sie die Spaltenüberschrift oder einen einzelnen Wert aus, und wählen Sie dann eine der Statistiken oder vordefinierten Perzentile aus, oder geben Sie eine benutzerdefinierte Perzentile ein (z. B. **`p95.45`**).
4. Wenn Sie den Zeitraum ändern möchten, wählen Sie die Registerkarte Graphed metrics aus. Wählen Sie die Spaltenüberschrift oder einen einzelnen Wert aus, und wählen Sie dann einen anderen Wert aus.
5. Wenn Sie eine horizontale Anmerkung hinzufügen möchten, wählen Sie Graph options (Diagrammoptionen) und anschließend Add horizontal annotation (Horizontale Anmerkung einfügen) aus.

   a. Geben Sie unter Label (Bezeichnung) eine Bezeichnung für die Anmerkung ein.

   b. Geben Sie unter Value (Wert) den Metrikwert ein, an dem die horizontale Anmerkung angezeigt wird.

   c. Geben Sie unter Fill an, ob die Füllschattierung bei der Anmerkung verwendet werden soll. Wählen Sie beispielsweise `Above` oder `Below` als entsprechend zu füllenden Bereich aus. Wenn Sie `Between` angeben, wird ein anderes `Value`-Feld angezeigt und der Bereich des Diagramms zwischen zwei Werten wird gefüllt.

   d. Geben Sie für Axis (Achse) an, ob sich die Nummern in `Value` auf die der linken Y-Achse oder der rechten Y-Achse zugewiesenen Metrik beziehen, wenn das Diagramm mehrere Metriken enthält.

      Sie können die Füllfarbe einer Anmerkung ändern, indem Sie das Farbquadrat in der linken Spalte neben der Anmerkung auswählen.

   Wiederholen Sie die Schritte, um einem Diagramm mehrere horizontale Anmerkungen hinzuzufügen.

Um eine Anmerkung auszublenden, deaktivieren Sie das Kontrollkästchen in der linken Spalte für diese Anmerkung.

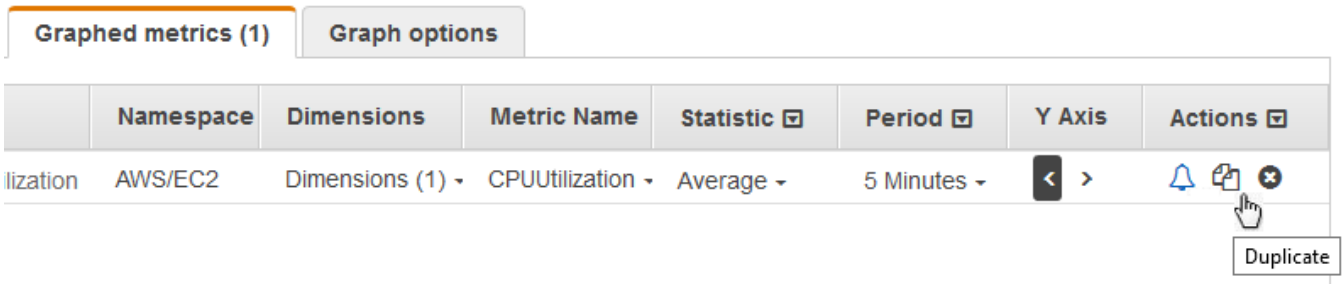
Um eine Anmerkung zu löschen, wählen Sie das x in der Spalte Actions aus.

6. Wählen Sie zum Ändern des Aktualisierungsintervalls Refresh options (Optionen aktualisieren) und anschließend Auto refresh (Automatisch aktualisieren) aus, oder wählen Sie 1 Minute (Eine Minute), 2 Minutes (2 Minuten), 5 Minutes (5 Minuten) oder 15 Minutes (15 Minuten) aus.

Duplizieren einer Metrik

So duplizieren Sie eine Metrik

1. Wählen Sie die Registerkarte Graphed metrics (Grafisch dargestellte Metriken) aus.
2. Wählen Sie für Actions das Symbol Duplicate aus.



3. Aktualisieren Sie die duplizierte Metrik bei Bedarf.

Zwei Diagramme zu einem zusammenführen

Sie können zwei verschiedene Diagramme zu einem zusammenführen. Das resultierende Diagramm zeigt dann beide Metriken. Dies kann nützlich sein, wenn Sie bereits verschiedene Metriken in verschiedenen Diagrammen anzeigen lassen und diese kombinieren möchten, oder wenn Sie auf einfache Weise ein einziges Diagramm mit Metriken aus verschiedenen Regionen erstellen möchten.

Um ein Diagramm in ein anderes einzubinden, verwenden Sie entweder die URL oder die JSON-Quelle des Diagramms, das Sie einbinden möchten.

Um zwei Diagramme zu einem zusammenzuführen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

2. Öffnen Sie das Diagramm, das Sie mit einem anderen Diagramm zusammenführen möchten. Dazu können Sie Metriken, Alle Metriken auswählen und dann eine Metrik wählen, die grafisch dargestellt werden soll. Sie können auch ein Dashboard öffnen und dann eines der Diagramme auf dem Dashboard öffnen, indem Sie das Diagramm auswählen und im Menü oben rechts im Diagramm In Metriken öffnen wählen.
3. Nachdem Sie ein Diagramm geöffnet haben, führen Sie einen der folgenden Schritte aus:
   - Kopieren Sie die URL aus der Browserleiste.
   - Wählen Sie die Registerkarte Quelle und dann Kopieren aus.
4. Öffnen Sie das Diagramm, mit dem Sie das vorherige Diagramm zusammenführen möchten.
5. Wenn Sie das zweite Diagramm in der Metrikansicht geöffnet haben, wählen Sie Aktionen, Diagramm zusammenführen aus.
6. Geben Sie die URL oder JSON ein, die Sie zuvor kopiert haben, und wählen Sie Zusammenführen.
7. Die zusammengeführten Diagramme werden angezeigt. Die y-Achse auf der linken Seite ist für das ursprüngliche Diagramm und die y-Achse auf der rechten Seite ist für das Diagramm, das Sie in das Diagramm eingefügt haben.

   > **Note**
   >
   > Wenn das Diagramm, in das Sie eingefügt haben, die Funktion METRICS() verwendet, werden die Metriken des eingefügten Diagramms nicht in die METRICS()-Berechnung des eingefügten Diagramms einbezogen.

8. Um das zusammengeführte Diagramm in einem Dashboard zu speichern, wählen Sie, Aktionen und Zum Dashboard hinzufügen aus.

Dynamische Labels verwenden

Sie können dynamische Beschriftungen für Ihre Diagramme verwenden. Dynamische Beschriftungen fügen dem Beschriftungsfeld für die ausgewählte Metrik einen dynamisch aktualisierten Wert hinzu. Sie können den Bezeichnungen eine Vielzahl von Werten hinzufügen, wie in den folgenden Tabellen gezeigt.

Der in der Beschriftung angezeigte dynamische Wert ergibt sich aus dem aktuell im Diagramm angezeigten Zeitbereich. Der dynamische Teil der Bezeichnung wird automatisch aktualisiert, wenn entweder das Dashboard oder das Diagramm aktualisiert wird.

Wenn Sie eine dynamische Beschriftung mit einem Suchbegriff verwenden, gilt die dynamische Beschriftung für jede von der Suche zurückgegebene Metrik.

Sie können die CloudWatch Konsole verwenden, um einem Label einen dynamischen Wert hinzuzufügen, das Label zu bearbeiten, die Position des dynamischen Werts innerhalb der Labelspalte zu ändern und andere Anpassungen vorzunehmen.

Dynamische Bezeichnungen

Innerhalb einer dynamischen Bezeichnung können Sie die folgenden Werte in Bezug auf die Eigenschaften der Metrik verwenden:

Live-Wert des dynamischen Labels	Beschreibung
${AVG}	Der Mittelwert der Werte im Zeitbereich, der derzeit im Diagramm angezeigt wird.
${DATAPOINT_COUNT}	Die Anzahl der Datenpunkte im Zeitbereich, der derzeit im Diagramm angezeigt wird.
${FIRST}	Der älteste der Metrikwerte im Zeitbereich, der aktuell in der Grafik angezeigt wird.
${FIRST_LAST_RANGE}	Der Unterschied zwischen den Metrikwerten der ältesten und neuesten Datenpunkte, die derzeit im Diagramm angezeigt werden.
${FIRST_LAST_TIME_RANGE}	Der absolute Zeitbereich zwischen den ältesten und neuesten Datenpunkten, die derzeit im Diagramm angezeigt werden.
${FIRST_TIME}	Der Zeitstempel des ältesten Datenpunkts im Zeitbereich, der aktuell im Diagramm angezeigt wird.
${FIRST_TIME_RELATIVE}	Die absolute Zeitdifferenz zwischen jetzt und dem Zeitstempel des ältesten Datenpunkts im Zeitbereich, der aktuell in der Grafik angezeigt wird.
${LABEL}	Die Darstellung der Standardbeschriftung für eine Metrik.

Live-Wert des dynamischen Labels	Beschreibung
${LAST}	Der aktuellste Messwert im Zeitbereich, der aktuell in der Grafik angezeigt wird.
${LAST_TIME}	Der Zeitstempel des neuesten Datenpunkts im Zeitbereich, der aktuell im Diagramm angezeigt wird.
${LAST_TIME_RELATIVE}	Die absolute Zeitdifferenz zwischen jetzt und dem Zeitstempel des neuesten Datenpunkts im Zeitbereich, der aktuell in der Grafik angezeigt wird.
${MAX}	Das Maximum der Werte im Zeitbereich, der derzeit im Diagramm angezeigt wird.
$ {MAX_TIME}	Der Zeitstempel des Datenpunkts mit dem höchsten Metrikwert der Datenpunkte, die derzeit im Diagramm angezeigt werden.
${MAX_TIME_RELATIVE}	Die absolute Zeitdifferenz zwischen jetzt und dem Zeitstempel des Datenpunkts mit dem höchsten Wert der Datenpunkte, die derzeit im Diagramm angezeigt werden.
${MIN}	Das Minimum der Werte im Zeitbereich, der derzeit im Diagramm angezeigt wird.
${MIN_MAX_RANGE}	Die Differenz der Metrikwerte zwischen den Datenpunkten mit den höchsten und niedrigsten Metrikwerten der Datenpunkte, die derzeit im Diagramm angezeigt werden.
${MIN_MAX_TIME_RANGE}	Der absolute Zeitbereich zwischen den Datenpunkten mit den höchsten und niedrigsten metrischen Werten der Datenpunkte, die derzeit im Diagramm angezeigt werden.
${MIN_TIME}	Der Zeitstempel des Datenpunkts mit dem niedrigsten Metrikwert der Datenpunkte, die derzeit im Diagramm angezeigt werden.

Live-Wert des dynamischen Labels	Beschreibung
${MIN_TIME_RELATIVE}	Die absolute Zeitdifferenz zwischen jetzt und dem Zeitstempel des Datenpunkts mit dem niedrigsten Wert der Datenpunkte, die derzeit im Diagramm angezeigt werden.
$ {PROP ('AccountId')}	Die AWS Konto-ID der Metrik.
$ {PROP ('AccountLabel')}	Die Bezeichnung, die für das Quellkonto angegeben wurde, dem diese Metrik gehört. Dabei handelt es sich um CloudWatch kontenübergreifende Beobachtbarkeit.
${PROP('Dim.*dimension_name*')}	Der Wert der angegebenen Dimension. Ersetzen Sie *dimension_name* mit dem Namen Ihrer Dimension, bei dem Groß- und Kleinschreibung beachtet wird.
$ {PROP ('')} MetricName	Name der Metrik.
${PROP('Namespace')}	Der Namespace der Metrik.
${PROP('Period')}	Der Zeitraum der Metrik in Sekunden.
${PROP('Region')}	Die AWS Region, in der die Metrik veröffentlicht wird.
${PROP('Stat')}	Die Metrik-Statistik, die grafisch dargestellt wird.
${SUM}	Die Summe der Werte im Zeitbereich, der derzeit im Diagramm angezeigt wird.

Angenommen, Sie haben einen Suchbegriff **`SEARCH(' {AWS/Lambda, FunctionName} Errors ', 'Sum')`**, der die `Errors` für jede Ihrer Lambda-Funktionen findet. Wenn Sie die Beschriftung auf `[max: ${MAX} Errors for Function Name ${LABEL}]` festlegen, lautet die Beschriftung für jede Metrik [max: ***number*** Errors for Function Name ***Name***].

Sie können einem Label bis zu sechs dynamische Werte hinzufügen. Sie können den `${LABEL}`-Platzhalter nur einmal innerhalb jeder Beschriftung verwenden.

Den Zeitraum oder das Zeitzonenformat eines Diagramms ändern

In diesem Abschnitt wird beschrieben, wie Sie das Datum, die Uhrzeit und das Zeitzonenformat in einem CloudWatch Metrikdiagramm ändern können. Es beschreibt auch, wie Sie ein Diagramm vergrößern können, um einen bestimmten Zeitraum anzuwenden. Weitere Informationen zum Erstellen eines Diagramms finden Sie unter Grafisches Darstellen von Metriken.

> **Note**
>
> Wenn der Zeitraum eines Dashboards kürzer ist als der Zeitraum, der für ein Diagramm auf dem Dashboard verwendet wurde, tritt Folgendes ein:
>
> - Das Diagramm wird so geändert, dass es die Datenmenge anzeigt, die einem vollständigen Zeitraum für dieses Widget entspricht, auch wenn dieser länger als der Dashboard-Zeitraum ist. Dadurch wird sichergestellt, dass es mindestens einen Datenpunkt im Diagramm gibt.
> - Die Startzeit des Zeitraums für diesen Datenpunkt wird nach hinten angepasst, um sicherzustellen, dass mindestens ein Datenpunkt angezeigt werden kann.

Einen relativen Zeitraum festlegen

New interface

So geben Sie eine relative Zeitspanne für ein Diagramm an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken) und dann All metrics (Alle Metriken) aus. In der oberen rechten Ecke des Bildschirms können Sie einen der vordefinierten Zeiträume von 1 Stunde bis zu 1 Woche auswählen (1 Std., 3 Std., 12 Std., 1 T.,3 Tg., oder 1 W.). Alternativ können Sie Custom (Benutzerdefiniert) auswählen, um Ihren eigenen Zeitraum festzulegen.
3. Wählen Sie Custom (Benutzerdefiniert) und dann die Registerkarte Relative (Relativ) in der oberen linken Ecke des Felds aus. Sie können einen Zeitraum in Minutes (Minuten), Hours (Stunden), Days (Tagen), Weeks (Wochen) oder Months (Monaten) festlegen.
4. Wählen Sie nach dem Festlegen eines Zeitraums Apply (Anwenden) aus.

Original interface

So geben Sie eine relative Zeitspanne für ein Diagramm an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken) und dann All metrics (Alle Metriken) aus. In der oberen rechten Ecke des Bildschirms können Sie einen der vordefinierten Zeiträume von 1 Stunde bis zu 1 Woche auswählen (1 Std., 3 Std., 12 Std., 1 T.,3 Tg., oder 1 W.). Alternativ können Sie Custom (Benutzerdefiniert) auswählen, um Ihren eigenen Zeitraum festzulegen.
3. Wählen Sie Custom (Benutzerdefiniert) und dann Relative (Relativ) in der oberen linken Ecke des Felds aus. Sie können einen Zeitraum in Minutes (Minuten), Hours (Stunden), Days (Tagen), Weeks (Wochen) oder Months (Monaten) festlegen.

Einen absoluten Zeitraum festlegen

New interface

So legen Sie eine absolute Zeitspanne für ein Diagramm fest

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken) und dann All metrics (Alle Metriken) aus. In der oberen rechten Ecke des Bildschirms können Sie einen der vordefinierten Zeiträume von 1 Stunde bis zu 1 Woche auswählen (1 Std., 3 Std., 12 Std., 1 T.,3 Tg., oder 1 W.). Alternativ können Sie Custom (Benutzerdefiniert) auswählen, um Ihren eigenen Zeitraum festzulegen.
3. Wählen Sie Custom (Benutzerdefiniert) und dann die Registerkarte Absolute (Absolut) in der oberen linken Ecke des Felds aus. Verwenden Sie die Kalenderauswahl oder die Textfelder, um einen Zeitraum festzulegen.
4. Wählen Sie nach dem Festlegen eines Zeitraums Apply (Anwenden) aus.

Original interface

So legen Sie eine absolute Zeitspanne für ein Diagramm fest

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

2. Wählen Sie im Navigationsbereich Metrics (Metriken) und dann All metrics (Alle Metriken) aus. In der oberen rechten Ecke des Bildschirms können Sie einen der vordefinierten Zeiträume von 1 Stunde bis zu 1 Woche auswählen (1 Std., 3 Std., 12 Std., 1 T.,3 Tg., oder 1 W.). Alternativ können Sie Custom (Benutzerdefiniert) auswählen, um Ihren eigenen Zeitraum festzulegen.
3. Wählen Sie Custom (Benutzerdefiniert) und dann Absolute (Absolut) in der oberen linken Ecke des Felds aus. Verwenden Sie die Kalenderauswahl oder die Textfelder, um einen Zeitraum festzulegen.
4. Wählen Sie nach dem Festlegen eines Zeitraums Apply (Anwenden) aus.

Ein Zeitzonenformat festlegen

New interface

Die Zeitzone für ein Diagramm geben Sie wie folgt an:

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken) und dann All metrics (Alle Metriken) aus. In der oberen rechten Ecke des Bildschirms können Sie einen der vordefinierten Zeiträume von 1 Stunde bis zu 1 Woche auswählen (1 Std., 3 Std., 12 Std., 1 T.,3 Tg., oder 1 W.). Alternativ können Sie Custom (Benutzerdefiniert) auswählen, um Ihren eigenen Zeitraum festzulegen.
3. Wählen Sie Custom (Benutzerdefiniert) und dann das Dropdown-Menü in der oberen rechten Ecke des Felds. Sie können die Zeitzone auf UTC oder Local time zone (lokale Zeitzone) ändern.
4. Nachdem Sie Ihre Änderungen vorgenommen haben, wählen Sie Apply (Anwenden) aus.

Original interface

Die Zeitzone für ein Diagramm geben Sie wie folgt an:

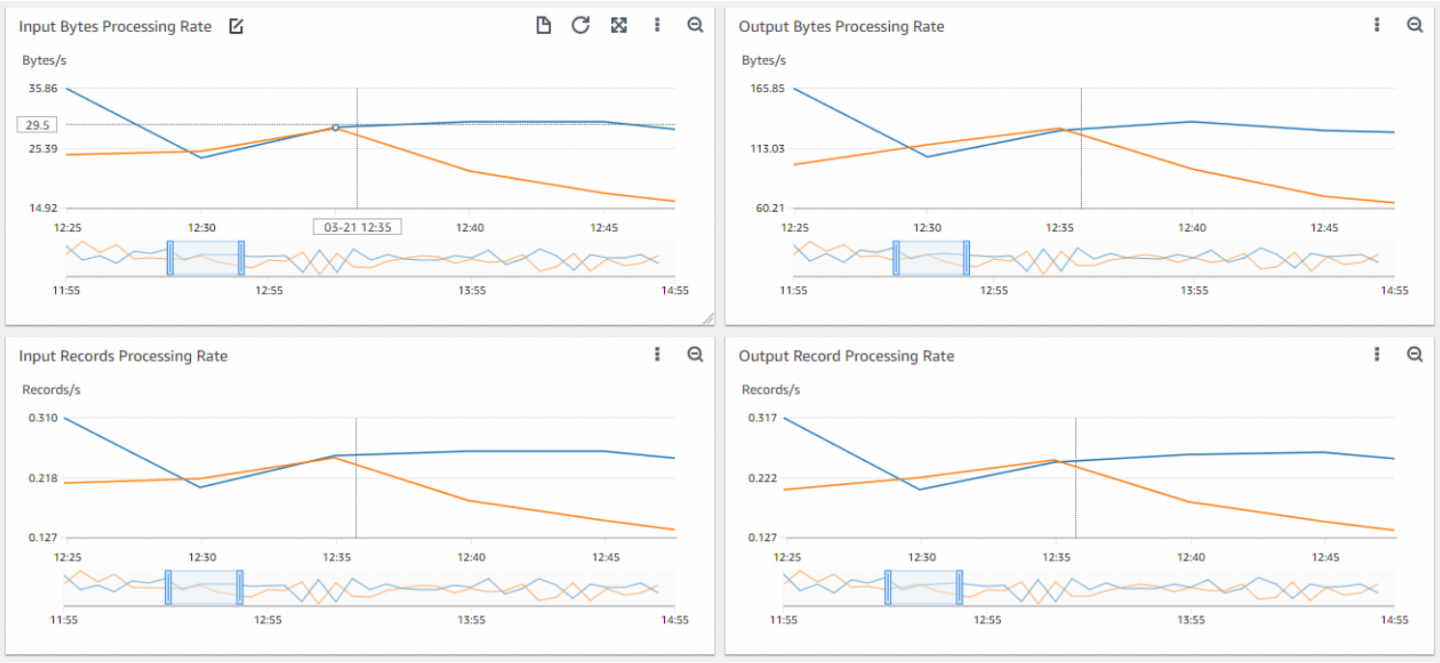
1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken) und dann All metrics (Alle Metriken) aus. In der oberen rechten Ecke des Bildschirms können Sie einen der vordefinierten Zeiträume von 1 Stunde bis zu 1 Woche auswählen (1 Std., 3 Std., 12 Std., 1 T.,3 Tg.,

oder 1 W.). Alternativ können Sie Custom (Benutzerdefiniert) auswählen, um Ihren eigenen Zeitraum festzulegen.

3. Wählen Sie Custom (Benutzerdefiniert) und dann das Dropdown-Menü in der oberen rechten Ecke des Felds. Sie können die Zeitzone auf UTC oder lokale Zeitzone ändern.

In ein Liniendiagramm oder ein gestapeltes Flächendiagramm hineinzoomen

In der CloudWatch Konsole können Sie die Minikap-Zoom-Funktion verwenden, um sich auf Abschnitte von Liniendiagrammen und gestapelten Flächendiagrammen zu konzentrieren, ohne zwischen der vergrößerten und der verkleinerten Ansicht wechseln zu müssen. Sie können beispielsweise das Mini-Map-Zoom-Feature verwenden, um eine Spitze in einem Liniendiagramm zu markieren, so dass Sie die Spitze mit anderen Metriken in Ihrem Dashboard auf derselben Zeitachse vergleichen können. In diesem Abschnitt wird beschrieben, wie Sie das Zoom-Feature verwenden können.



Im vorstehenden Image konzentriert sich das Zoom-Feature auf eine Spitze in einem Liniendiagramm, die mit der Verarbeitungsrate der eingegebenen Bytes zusammenhängt, während gleichzeitig andere Liniendiagramme im Dashboard angezeigt werden, die sich auf Abschnitte derselben Zeitleiste konzentrieren.

New interface

So vergrößern Sie ein Diagramm

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken) und dann All metrics (Alle Metriken) aus.
3. Wählen Sie Browse (Durchsuchen) und wählen Sie dann eine oder mehrere Metriken aus, die Sie grafisch darstellen möchten.
4. Wählen Sie Options (Optionen) und wählen Sie Line (Linie) unter Widget type (Widget-Typ) aus.
5. Wählen Sie den Bereich des Diagramms aus, auf den Sie sich konzentrieren möchten, und lassen Sie die Maustaste los.
6. Um den Zoom zurückzusetzen, wählen Sie das Symbol Zoom zurücksetzen aus, das wie eine Lupe mit einem Minuszeichen (-) darin aussieht.

Original interface

So vergrößern Sie ein Diagramm

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken) und dann All metrics (Alle Metriken).
3. Wählen Sie All metrics (Alle Metriken), und wählen Sie dann eine Metrik aus, die Sie grafisch darstellen möchten.
4. Wählen Sie Graph options (Diagrammoptionen). Wählen Sie unter Widget type (Widget-Typ) die Option Line (Linie).
5. Wählen Sie den Bereich des Diagramms aus, auf den Sie sich konzentrieren möchten, und lassen Sie die Maustaste los.
6. Um den Zoom zurückzusetzen, wählen Sie das Symbol Zoom zurücksetzen aus, das wie eine Lupe mit einem Minuszeichen (-) darin aussieht.

> **Tip**
>
> Wenn Sie bereits ein Dashboard erstellt haben, das ein Liniendiagramm oder ein gestapeltes Flächendiagramm enthält, können Sie zum Dashboard wechseln und das Zoom-Feature verwenden.

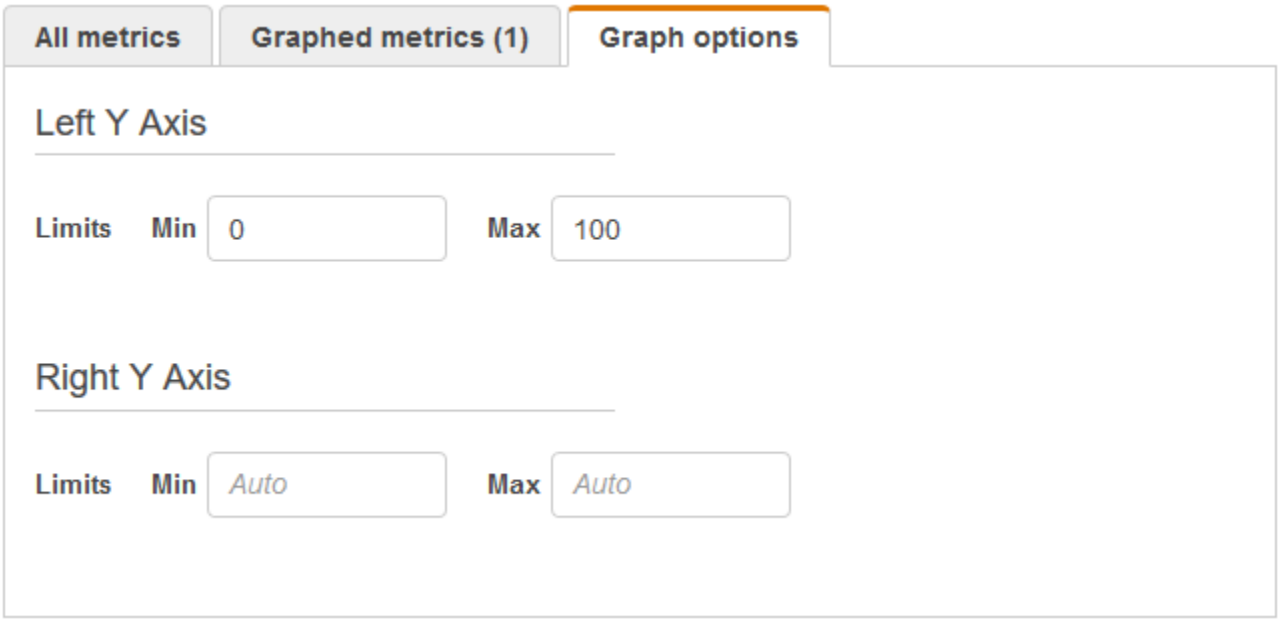
Die y-Achse in einem Diagramm ändern

Sie können benutzerdefinierte Grenzen für die Y-Achse in einem Diagramm festlegen, um die Daten genauer anzuzeigen. Sie können beispielsweise die Grenzen in einem `CPUUtilization`-Diagramm auf 100 % ändern, damit einfacher zu erkennen ist, ob die CPU-Auslastung niedrig (die gezeichnete Linie befindet sich am unteren Rand des Diagramms) oder hoch (die gezeichnete Linie befindet sich am oberen Rand des Diagramms) ist.

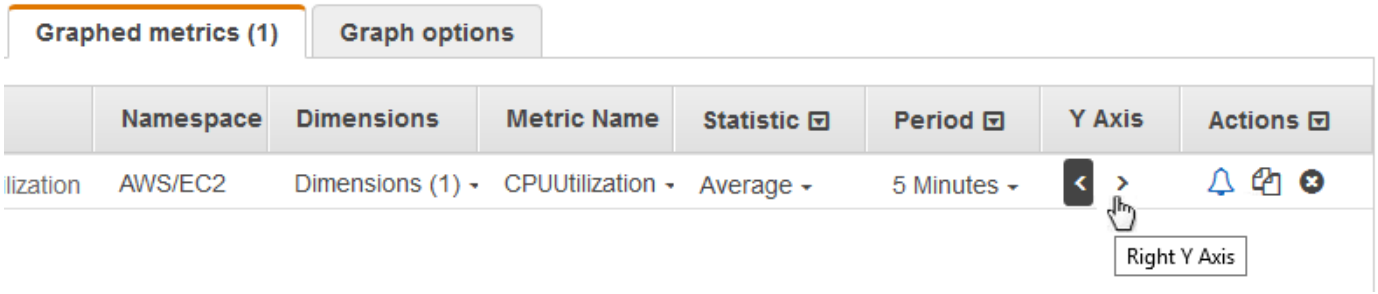
Sie können zwischen zwei verschiedenen Y-Achsen für Ihr Diagramm wählen. Dies ist nützlich, wenn das Diagramm Metriken enthält, die unterschiedliche Einheiten haben oder deren Wertebereich sehr unterschiedlich ist.

So ändern Sie die Y-Achse in einem Diagramm

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie einen Metrik-Namespace (z. B. EC2) und dann eine Metrik-Dimension (z. B. Per-Instance Metrics (Metriken pro Instance)) aus.
4. Die Registerkarte All metrics zeigt alle Metriken für diese Dimension in diesem Namespace an. Um eine Metrik grafisch darzustellen, müssen Sie das Kontrollkästchen neben der Metrik aktivieren.
5. Geben Sie auf der Registerkarte Graph options die Min und Max-Werte für Left Y Axis ein. Der Wert für Min darf nicht größer sein als der Wert für Max.



6. Zum Erstellen einer zweiten Y-Achse geben Sie die Min- und Max-Werte für Right Y Axis (Rechte Y-Achse) ein.
7. Zum Wechseln zwischen den beiden Y-Achsen wählen Sie die Registerkarte Graphed metrics (Grafisch dargestellte Metriken) aus. Wählen Sie für Y Axis die Option Left Y Axis oder Right Y Axis aus.



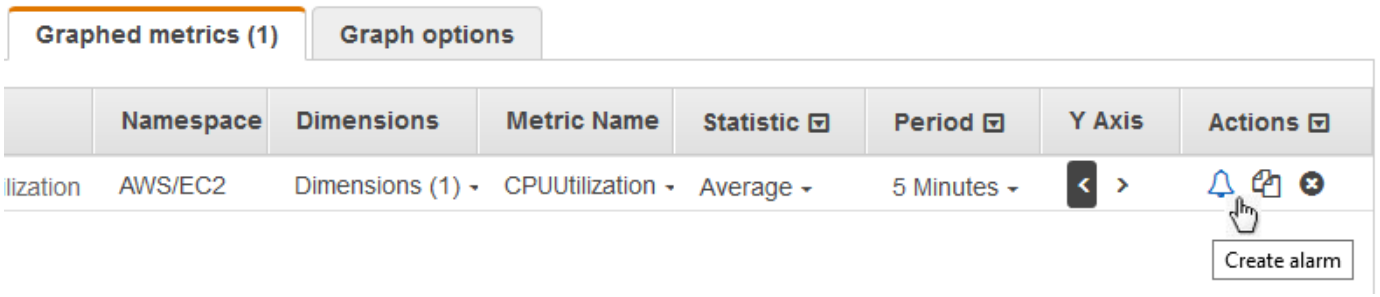
Einen Alarm aus einer Metrik in einem Diagramm erstellen

Sie können eine Metrik in einem Diagramm darstellen und aus der Metrik einen Alarm in dem Diagramm einrichten. Das hat den Vorteil, dass viele Alarmfelder für Sie ausgefüllt werden.

So erstellen Sie einen Alarm aus einer Metrik in einem Diagramm

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.

3. Wählen Sie einen Metrik-Namespace (z. B. EC2) und dann eine Metrik-Dimension (z. B. Per-Instance Metrics (Metriken pro Instance)) aus.
4. Die Registerkarte All metrics zeigt alle Metriken für diese Dimension in diesem Namespace an. Um eine Metrik grafisch darzustellen, müssen Sie das Kontrollkästchen neben der Metrik aktivieren.
5. Wenn Sie einen Alarm für die Metrik erstellen möchten, wählen Sie die Registerkarte Graphed metrics aus. Wählen Sie für Actions das Alarm-Symbol aus.



6. Wählen Sie in Conditions (Bedingungen) die Optionen Static (Statisch) oder Anomaly detection (Anomalieerkennung) aus, um anzugeben, ob ein statischer Schwellenwert oder ein Anomalieerkennungsmodell für den Alarm verwendet werden soll.

   Geben Sie abhängig von Ihrer Entscheidung die restlichen Daten für die Alarmbedingungen ein.
7. Wählen Sie Additional configuration (Zusätzliche Konfiguration). Geben Sie unter Datapoints to alarm (Datenpunkte für Alarm) an, wie viele Auswertungszeiträume (Datenpunkte) im Status `ALARM` sein müssen, damit der Alarm ausgelöst wird. Wenn die beiden Werte hier übereinstimmen, erstellen Sie einen Alarm, der in den Status `ALARM` wechselt, wenn entsprechend viele aufeinanderfolgende Zeiträume überschritten werden.

   Um einen M aus N Alarm zu erstellen, geben Sie eine niedrigere Zahl für den ersten Wert als für den zweiten Wert an. Weitere Informationen finden Sie unter Auswerten eines Alarms.
8. Wählen Sie für Missing data treatment (Behandlung von fehlenden Daten) aus, wie sich der Alarm verhalten soll, wenn einige Datenpunkte fehlen. Weitere Informationen finden Sie unter Konfiguration, wie Alarme fehlende Daten behandeln CloudWatch .
9. Wählen Sie Weiter.
10. Wählen Sie unter Notification (Benachrichtigung) ein SNS-Thema aus, das benachrichtigt werden soll, wenn sich der Alarm im Status `ALARM`, `OK` oder `INSUFFICIENT_DATA` befindet.

    Um zu erreichen, dass der Alarm mehrere Benachrichtigungen für den gleichen Alarmstatus oder für verschiedene Statuswerte sendet, wählen Sie Benachrichtigung hinzufügen.

Damit der Alarm keine Benachrichtigungen sendet, wählen Sie Remove (Entfernen).

11. Um den Alarm Auto-Scaling- oder EC2-Aktionen durchführen zu lassen, wählen Sie die entsprechende Schaltfläche und wählen Sie den Alarmstatus und die auszuführende Aktion.
12. Wenn Sie fertig sind, wählen Sie Weiter.
13. Geben Sie einen Namen und eine Beschreibung für den Alarm ein. Der Name darf nur ASCII-Zeichen enthalten. Wählen Sie anschließend Weiter.
14. Bestätigen Sie unter Preview and create (Vorschau und erstellen), dass die Informationen und Bedingungen den Anforderungen entsprechen, und wählen Sie dann Create alarm (Alarm erstellen).

Verwendung der CloudWatch Anomalieerkennung

Wenn Sie die Anomalieerkennung für eine Metrik aktivieren, CloudWatch wendet statistische Algorithmen und Algorithmen für maschinelles Lernen an. Diese Algorithmen analysieren kontinuierlich Metriken von Systemen und Anwendungen, ermitteln normale Baseline-Werte und zeigen Anomalien an, wobei nur minimale Benutzereingriffe erforderlich sind.

Die Algorithmen erzeugen ein Anomalieerkennungsmodell. Das Modell generiert einen Bereich von erwarteten Werten, die ein normales Metrikverhalten darstellen.

Sie können die Anomalieerkennung mithilfe des SDK AWS Management Console, des AWS CLI AWS CloudFormation, oder des AWS SDK aktivieren. Sie können die Anomalieerkennung für von angebotene Metriken AWS und auch für benutzerdefinierte Metriken aktivieren. In einem Konto, das als Überwachungskonto für CloudWatch kontenübergreifende Beobachtbarkeit eingerichtet wurde, können Sie zusätzlich zu den Metriken im Überwachungskonto Anomaliedetektoren für Kennzahlen in Quellkonten einrichten.

Sie können das Modell der erwarteten Werte auf zwei Arten einsetzen:

- Sie können Anomalieerkennungsalarme basierend auf dem erwarteten Wert einer Metrik erstellen. Diese Arten von Alarmen besitzen keinen statischen Schwellenwert für die Bestimmung des Alarmzustands. Stattdessen vergleichen sie den Wert der Metrik mit dem erwarteten Wert basierend auf dem Anomalieerkennungsmodell.

  Sie können festlegen, ob der Alarm ausgelöst werden soll, wenn der Metrikwert das Band erwarteter Werte überschreitet, unterschreitet oder sowohl über- als auch unterschreitet.

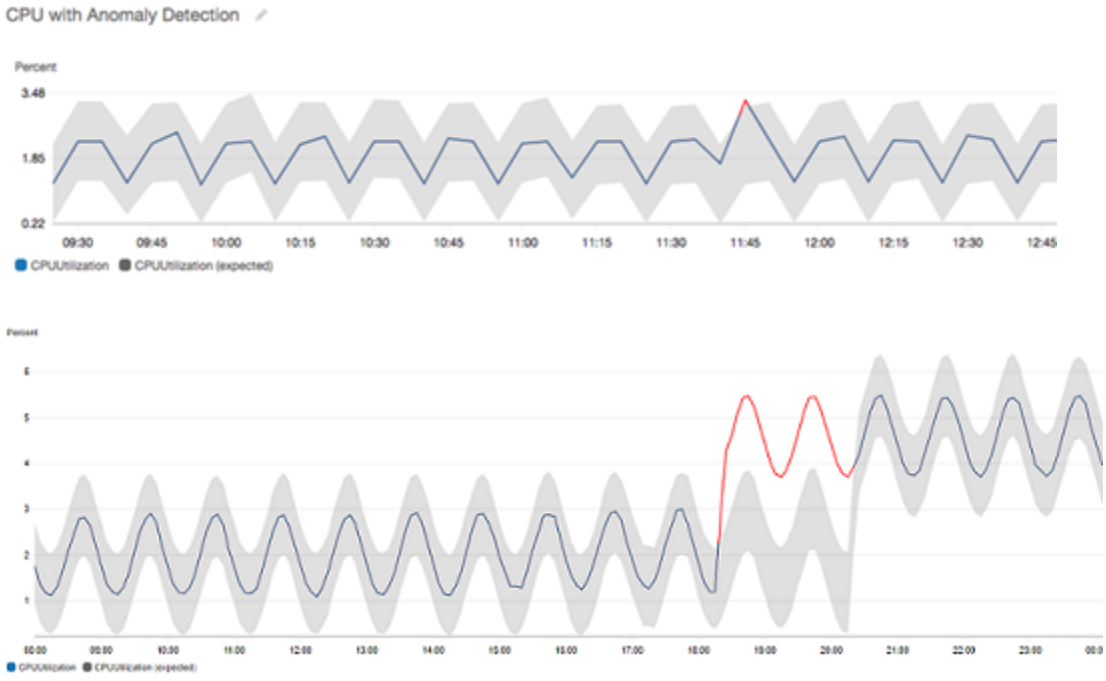
Weitere Informationen finden Sie unter Erstellen Sie einen CloudWatch Alarm, der auf der Erkennung von Anomalien basiert.

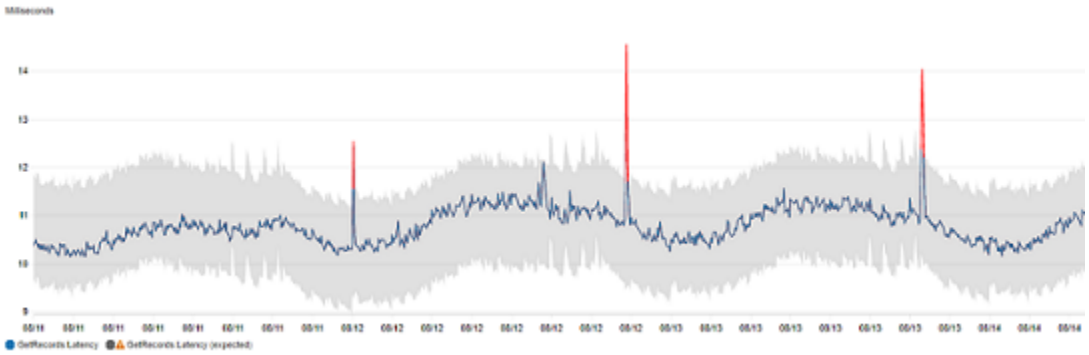
- Wenn Sie ein Diagramm mit Metrikdaten anzeigen, können Sie die erwarteten Werte im Diagramm als Band darstellen. Dies zeigt visuell, welche Werte im Diagramm außerhalb des normalen Bereichs liegen. Weitere Informationen finden Sie unter Erstellen eines Diagramms.

  Sie können die oberen und unteren Werte des Bands des Modells auch mithilfe der `GetMetricData`-API-Anforderung mit der mathematischen Funktion der Metrik `ANOMALY_DETECTION_BAND` abrufen. Weitere Informationen finden Sie unter. GetMetricData

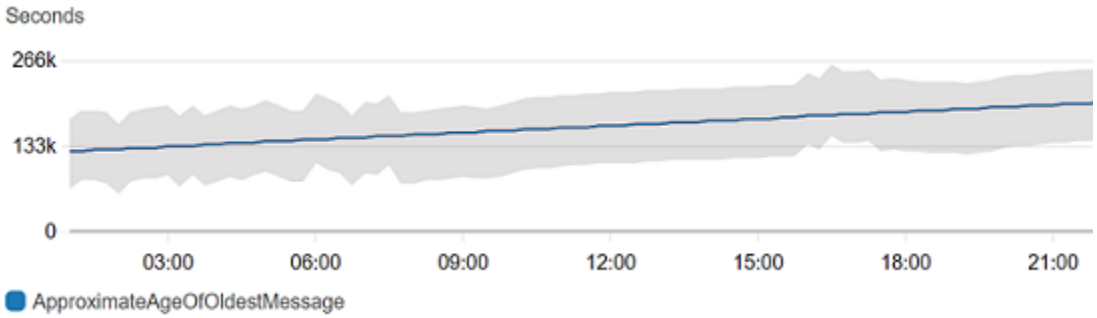
In einem Diagramm mit Anomalieerkennung wird der erwartete Wertebereich als graues Band angezeigt. Wenn der tatsächliche Wert der Metrik außerhalb dieses Bands liegt, wird er während dieser Zeit rot angezeigt.

Anomalieerkennungsalgorithmen berücksichtigen saisonale und trendbasierte Änderungen von Metriken. Die saisonalen Änderungen können stündlich, täglich oder wöchentlich erfolgen, wie in den folgenden Beispielen gezeigt.

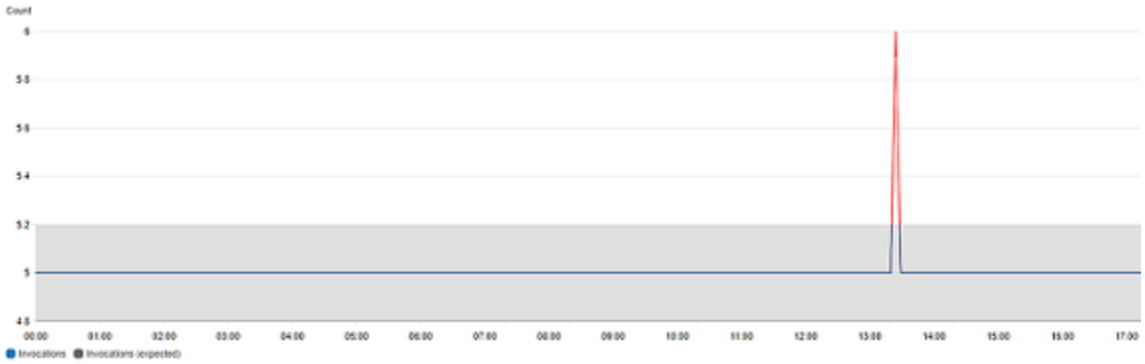




Die längerfristigen Trends könnten nach unten oder nach oben zeigen.



Die Anomalieerkennung funktioniert auch gut mit Metriken mit flachen Mustern.



So funktioniert die Erkennung von CloudWatch Anomalien

Wenn Sie die Anomalieerkennung für eine Metrik aktivieren, CloudWatch wendet Algorithmen für maschinelles Lernen auf die früheren Daten der Metrik an, um ein Modell der erwarteten Werte der Metrik zu erstellen. Das Modell bewertet sowohl Trends als auch stündliche, tägliche und wöchentliche Muster der Metrik. Der Algorithmus wird anhand von bis zu zwei Wochen Metrikdaten trainiert. Sie können die Anomalieerkennung für eine Metrik jedoch auch dann aktivieren, wenn es für die Metrik keine vollständigen zwei Wochen an Daten gibt.

Sie geben einen Wert für den Schwellenwert für die Erkennung von Anomalien an, der zusammen mit dem Modell CloudWatch verwendet wird, um den „normalen“ Wertebereich für die Metrik zu bestimmen. Ein höherer Wert für den Schwellenwert für die Anomalieerkennung führt zu einem breiteren Band „normaler“ Werte.

Das Machine Learning-Modell ist für eine Metrik und eine Statistik jeweils spezifisch. Beispiel: Wenn Sie die Anomalieerkennung für eine Metrik mit der `AVG`-Statistik aktivieren, ist das Modell spezifisch für die `AVG`-Statistik.

Wenn ein Modell für viele gängige Metriken von AWS Diensten CloudWatch erstellt wird, wird sichergestellt, dass das Band nicht über die logischen Werte hinausgeht. Beispielsweise bleibt das Band für `MemoryUtilization` eine EC2-Instance zwischen 0 und 100, und das Band-Tracking CloudFront `Requests`, das nicht negativ sein kann, wird sich niemals unter Null erstrecken.

Nachdem Sie ein Modell erstellt haben, bewertet die CloudWatch Anomalieerkennung das Modell kontinuierlich und nimmt Anpassungen vor, um sicherzustellen, dass es so genau wie möglich ist. Dies umfasst ein erneutes Training des Modells, um es anzupassen, wenn sich die Metrikwerte im Laufe der Zeit entwickeln oder plötzliche Änderungen aufweisen, und umfasst auch Prädiktoren zur Verbesserung der Modelle von Metriken, die saisonal, sehr hoch oder spärlich sind.

Nachdem Sie die Anomalieerkennung für eine Metrik aktiviert haben, können Sie festlegen, dass bestimmte Zeiträume der Metrik beim Trainieren des Modells nicht berücksichtigt werden sollen. Auf diese Weise können Sie Bereitstellungen oder andere ungewöhnliche Ereignisse aus dem Modelltraining ausschließen, damit ein möglichst genaues Modell erstellt wird.

Wenn Sie Modelle zur Erkennung von Anomalien für Alarme verwenden, fallen Gebühren auf Ihrem Konto an. AWS Weitere Informationen finden Sie unter Amazon CloudWatch – Preise.

Anomalieerkennung bei Metrikberechnungen

Die Anomalieerkennung bei Metrikberechnungen ist ein Feature, mit der Sie Alarme für die Anomalieerkennung auf Grundlage der Ausgaben der Metrikberechnungen erstellen können. Sie können diese Ausdrücke verwenden, um Diagramme zu erstellen, die Anomalieerkennungsbänder visualisieren. Das Feature unterstützt grundlegende arithmetische Funktionen, Vergleichs- und logische Operatoren sowie die meisten anderen Funktionen. Informationen zu Funktionen, die nicht unterstützt werden, finden Sie unter Using metric math im CloudWatch Amazon-Benutzerhandbuch.

Sie können Anomalieerkennungsmodelle basierend auf Ausdrücken der Metrikberechnung erstellen, ähnlich wie die Erstellung von Anomalieerkennungsmodellen. Von der CloudWatch Konsole aus

können Sie die Anomalieerkennung auf metrische mathematische Ausdrücke anwenden und die Anomalieerkennung als Schwellenwerttyp für diese Ausdrücke auswählen.

> **Note**
>
> Die Anomalieerkennung bei Metrikberechnungen kann nur in der neuesten Version der Metrik-Benutzeroberfläche aktiviert und bearbeitet werden. Wenn Sie Anomaliedetektoren basierend auf Ausdrücken der Metrikberechnung in der neuen Version der Schnittstelle erstellen, können Sie sie in der alten Version anzeigen, aber nicht bearbeiten.

Informationen zum Erstellen von Alarmen und Modellen für die Anomalieerkennung und Metrikberechnungen finden Sie in den folgenden Abschnitten:

- Erstellen eines CloudWatch Alarms auf der Grundlage der Anomalieerkennung
- Erstellen eines CloudWatch Alarms auf der Grundlage eines metrischen mathematischen Ausdrucks

Sie können mithilfe der API auch Modelle zur Erkennung von Anomalien erstellen, löschen und entdecken, die auf metrischen mathematischen Ausdrücken basieren, indem Sie die CloudWatch API mit `PutAnomalyDetectorDeleteAnomalyDetector`, und `DescribeAnomalyDetectors` verwenden. Informationen zu diesen API-Aktionen finden Sie in den folgenden Abschnitten der Amazon CloudWatch API-Referenz.

- PutAnomalyDetector
- DeleteAnomalyDetector
- DescribeAnomalyDetectors

Informationen zu den Preisen für Alarme zur Erkennung von Anomalien finden Sie unter CloudWatch Amazon-Preise.

Verwenden von Metrikberechnungen

Mit metrischer Mathematik können Sie mehrere CloudWatch Metriken abfragen und mithilfe mathematischer Ausdrücke neue Zeitreihen auf der Grundlage dieser Metriken erstellen. Sie können die resultierenden Zeitreihen auf der CloudWatch Konsole visualisieren und sie zu Dashboards

hinzufügen. Am Beispiel von AWS Lambda Metriken könnten Sie die `Errors` Metrik durch die `Invocations` Metrik dividieren, um eine Fehlerquote zu erhalten. Fügen Sie dann die resultierende Zeitreihe zu einem Diagramm auf Ihrem CloudWatch Dashboard hinzu.

Sie können Metrikberechnungen auch mithilfe der `GetMetricData`-API-Operation programmgesteuert durchführen. Weitere Informationen finden Sie unter GetMetricData.

Fügen Sie einem CloudWatch Diagramm einen mathematischen Ausdruck hinzu

Sie können einem Diagramm auf Ihrem CloudWatch Dashboard einen mathematischen Ausdruck hinzufügen. Jedes Diagramm ist auf die Nutzung von maximal 500 Metriken und Ausdrücke begrenzt. Daher können Sie einen mathematischen Ausdruck nur dann hinzufügen, wenn das Diagramm 499 oder weniger Metriken aufweist. Dies gilt auch dann, wenn nicht alle Metriken im Diagramm angezeigt werden.

So fügen Sie einem Diagramm einen mathematischen Ausdruck hinzu

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Erstellen oder bearbeiten Sie ein Diagramm. Es muss mindestens eine Metrik im Diagramm vorhanden sein.
3. Wählen Sie Graphed metrics (Grafisch dargestellte Metrik) aus.
4. Wählen Sie die Option Math expression (mathematischer Ausdruck), Start with empty expression (Mit leerem Ausdruck beginnen) aus. Eine neue Zeile wird für den Ausdruck angezeigt.
5. Geben Sie in der neuen Zeile unter der Spalte Details den mathematischen Ausdruck ein. In den Tabellen im Abschnitt Metriken Mathematische Syntax und Funktionen werden die Funktionen aufgeführt, die Sie in dem Ausdruck verwenden können.

   Um eine Metrik oder das Ergebnis eines anderen Ausdrucks als Teil der Formel für diesen Ausdruck zu nutzen, verwenden Sie den in der Spalte Id (ID) angezeigten Wert, z. B. m1+m2 oder e1-MIN(e1).

   Sie können den Wert von Id (ID) ändern. Er kann Ziffern, Buchstaben und Unterstriche enthalten und muss mit einem Kleinbuchstaben beginnen. Sie können das Diagramm leichter lesbar machen, indem Sie den ID-Wert umbenennen, zum Beispiel von m1 und m2 zu errors (Fehler) und requests (Anfragen).

> **Tip**
>
> Wählen Sie den Pfeil nach unten neben Math Expression (Mathematischer Ausdruck), um eine Liste der unterstützten Funktionen anzuzeigen, die Sie beim Erstellen des Ausdrucks verwenden können.

6. Geben Sie in der Spalte Label (Bezeichnung) des Ausdrucks einen Namen ein, der beschreibt, was der Ausdruck berechnet.

   Wenn das Ergebnis eines Ausdrucks ein Array von Zeitreihen ist, wird jede dieser Zeitreihen in dem Diagramm als eigene Zeile in einer eigenen Farbe angezeigt. Direkt unterhalb des Diagramms wird für jede Zeile im Diagramm eine Legende angezeigt. Für einen einzelnen Ausdruck, der mehrere Zeitreihen erstellt, sind die Legendenuntertitel für diese Zeitreihen im Format ***Ausdruck-Beschriftungsmetrik-Beschriftung***. Beispiel: Wenn das Diagramm eine Metrik mit der Beschriftung Errors (Fehler) und einem Ausdruck FILL(METRICS(), 0) mit der Beschriftung Filled With 0: (Gefüllt mit 0:) enthält, lautet eine Zeile in der Legende Filled With 0: Errors (Gefüllt mit 0: Fehler). Damit in der Legende nur die ursprünglichen Bezeichnungen der Metriken angezeigt werden, muss *`Expression Label (Ausdruck-Bezeichnung)`* leer sein.

   Wenn ein Ausdruck ein Array mit Zeitreihen in dem Diagramm erzeugt, können Sie die Farben der einzelnen Zeitreihen nicht ändern.
7. Nachdem Sie die gewünschten Ausdrücke hinzugefügt haben, können Sie optional das Diagramm vereinfachen, indem Sie einen Teil der ursprünglichen Metriken ausblenden. Um eine Metrik oder einen Ausdruck auszublenden, deaktivieren Sie das Kontrollkästchen links neben dem Feld Id (ID).

Syntax und Funktionen von Metrikberechnungen

In den folgenden Abschnitten werden die für Metrikberechnungen verfügbaren Funktionen erläutert. Alle Funktionen müssen in Großbuchstaben geschrieben werden (z. B. AVG). Einträge im Feld Id (ID) für alle Metriken und mathematischen Ausdrücke müssen dagegen mit einem Kleinbuchstaben beginnen.

Das Endergebnis jedes mathematischen Ausdrucks muss eine einzelne Zeitreihe oder ein Array von Zeitreihen sein. Einige Funktionen erzeugen eine skalare Zahl. Sie können diese Funktionen innerhalb einer größeren Funktion verwenden, die letztendlich eine Zeitreihe generiert. Wenn Sie beispielsweise den Durchschnitt (AVG) einer einzelnen Zeitreihe errechnen, wird dadurch eine

skalare Zahl generiert, sodass dies nicht das endgültige Ausdrucksergebnis sein kann. Aber Sie könnten es in der Funktion m1-AVG(m1) verwenden, um eine Zeitreihe der Differenz zwischen jedem einzelnen Datenpunkt und dem Mittelwert in der Zeitreihe anzuzeigen.

Datentypabkürzungen

Einige Funktionen sind nur für bestimmte Arten von Daten gültig. Die Abkürzungen in der folgenden Liste werden in den Tabellen der Funktionen verwendet, um die Datentypen darzustellen, die für jede Funktion unterstützt werden:

- S stellt eine skalare Zahl dar, wie etwa 2, -5 oder 50,25.
- TS ist eine Zeitreihe (eine Reihe von Werten für eine einzelne CloudWatch Metrik im Zeitverlauf): zum Beispiel die `CPUUtilization` Metrik der `i-1234567890abcdef0` letzten drei Tage.
- TS [] ist ein Array von Zeitreihen, z. B. die Zeitreihe für mehrere Metriken.
- String[] ist ein Array von Zeichenfolgen.

Die METRICS()-Funktion

Die METRICS()-Funktion gibt alle Metriken in der Anfrage zurück. Mathematische Ausdrücke sind nicht enthalten.

Sie können METRICS() innerhalb eines größeren Ausdrucks verwenden, der letztendlich eine einzige Zeitreihe oder ein Array mit Zeitreihen generiert. Beispielsweise gibt der Ausdruck SUM(METRICS()) eine Zeitreihe (TS) zurück, die die Summe der Werte aller grafisch dargestellten Metriken ist. METRICS()/100 gibt ein Array von Zeitreihen zurück. Jede Zeitreihe zeigt dabei einen Datenpunkt eines der Metriken geteilt durch 100 an.

Sie können die METRICS()-Funktion mit einer Zeichenfolge verwenden, um nur die grafisch dargestellten Metriken zurückzugeben, die diese Zeichenfolge in ihrem Feld Id (ID) enthalten. Beispielsweise gibt der Ausdruck SUM(METRICS("errors")) eine Zeitreihe zurück, die die Summe der Werte aller grafisch dargestellten Metriken ist, die in ihrem Feld Id (ID) "errors", also Fehler, haben. Sie können auch SUM([METRICS(“4xx”), METRICS(“5xx”)]) verwenden, um mehrere Zeichenfolgen abzugleichen.

Grundlegende arithmetische Funktionen

In der folgenden Tabelle sind die unterstützten grundlegenden arithmetischen Funktionen aufgeführt. Fehlende Werte in Zeitreihen werden als 0 (Null) behandelt. Wenn der Wert eines Datenpunkts dazu führt, dass eine Funktion versucht, durch Null zu teilen, wird dieser Datenpunkt verworfen.

Operation	Argumente	Beispiele
Arithmetische Operatoren: + - * / ^	S, S	PERIOD(m1)/60
	S, TS	5 * m1
	TS, TS	m1 - m2
	S, TS[]	SUM(100/[m1, m2])
	TS, TS[]	AVG(METRICS())
		METRICS()*100
Unäre Subtraktion –	S	-5*m1
	TS	-m1
	TS[]	SUM(-[m1, m2])

Vergleichs- und logische Operatoren

Sie können Vergleichs- und logische Operatoren entweder mit einem Paar von Zeitreihen oder einem Paar einzelner Skalarwerte verwenden. Wenn Sie einen Vergleichsoperator mit einem Paar von Zeitreihen verwenden, geben die Operatoren eine Zeitreihe zurück, bei der jeder Datenpunkt entweder 0 (false) oder 1 (true) ist. Wenn Sie einen Vergleichsoperator für ein Paar von skalaren Werten verwenden, wird ein einzelner skalarer Wert zurückgegeben, entweder 0 oder 1.

Wenn Vergleichsoperatoren zwischen zwei Zeitreihen verwendet werden und nur eine der Zeitreihen einen Wert für einen bestimmten Zeitstempel aufweist, behandelt die Funktion den fehlenden Wert in der anderen Zeitreihe als 0.

Sie können logische Operatoren in Verbindung mit Vergleichsoperatoren verwenden, um komplexere Funktionen zu erstellen.

In der folgenden Tabelle sind die unterstützten Operatoren aufgeführt.

Art des Operators	Unterstützte Operatoren
Vergleichsoperatoren	==

Art des Operators	Unterstützte Operatoren
	!= <= >= < >
Logische Operatoren	UND oder && ODER oder ||

Um zu sehen, wie diese Operatoren verwendet werden, nehmen wir an, wir haben zwei Zeitreihen: metric1 hat Werte von `[30, 20, 0, 0]` und metric2 hat Werte von `[20, -, 20, -]`, wobei `-` anzeigt, dass es keinen Wert für diesen Zeitstempel gibt.

Expression	Output
(metric1 < metric2)	0, 0, 1, 0
(metric1 >= 30)	1, 0, 0, 0
(metric1 > 15 AND metric2 > 15)	1, 0, 0, 0

Für Metrikberechnungen unterstützte Funktionen

In der folgenden Tabelle werden die Funktionen beschrieben, die Sie in mathematischen Ausdrücken verwenden können. Geben Sie alle Funktionen in Großbuchstaben ein.

Das Endergebnis jedes mathematischen Ausdrucks muss eine einzelne Zeitreihe oder ein Array von Zeitreihen sein. Einige Funktionen in Tabellen in den folgenden Abschnitten erzeugen skalare Zahlen. Sie können diese Funktionen innerhalb einer größeren Funktion verwenden, die letztendlich eine Zeitreihe generiert. Wenn Sie beispielsweise den Durchschnitt (AVG) einer einzelnen Zeitreihe errechnen, wird dadurch eine skalare Zahl generiert, sodass dies nicht das endgültige Ausdrucksergebnis sein kann. Aber Sie könnten es in der Funktion m1-AVG(m1) verwenden,

um eine Zeitreihe der Differenz zwischen den einzelnen individuellen Datenpunkten und dem Durchschnittswert dieser Datenpunkte anzuzeigen.

In der folgenden Tabelle ist jedes Beispiel in der Spalte Examples (Beispiele) ein Ausdruck, der eine einzelne Zeitreihe oder ein Array von Zeitreihen erzeugt. Dies zeigt, wie Funktionen, die skalare Zahlen zurückgeben, in einem gültigen Ausdruck verwendet werden können, der eine einzelne Zeitreihe erzeugt.

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
ABS	TS TS[]	TS TS[]	Gibt den absoluten Wert des einzelnen Datenpunkts zurück.	ABS(m1-m2) MIN(ABS([m1, m2])) ABS(METRICS())	✓
ANOMALY_D ETECTION_ BAND	TS TS, S	TS[]	Gibt ein Anomaliee rkennungsband für die angegebene Metrik zurück. Das Band besteht aus zwei Zeitreihen, einer für die Obergrenze des "normalen" erwartete n Wertes der Metrik und die andere für die Untergrenze. Die Funktion besitzt zwei mögliche Argumente. Das erste ist die ID der Metrik, für die das Band erstellt werden soll. Das zweite Argument ist die Anzahl der für das Band zu verwenden	ANOMALY_D ETECTION_BAND(m1) ANOMALY_D ETECTION_BAND(m1,4)	

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			den Standarda bweichungen. Wenn Sie dieses Argument nicht angeben, wird der Standardwert 2 verwendet wird. Weitere Informationen finden Sie unter Verwendung der CloudWatch Anomaliee rkennung.		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
AVG	TS TS[]	S TS	Der Durchschnitt (AVG) einer einzelnen Zeitreihe gibt eine Skalarzahl zurück, die den Durchschn itt aller Datenpunkte in der Metrik darstellt . Der Durchschnitt (AVG) eines Arrays von Zeitreihen gibt eine einzige Zeitreihe zurück. Fehlende Werte werden als 0 (Null) behandelt. **Note** Wir empfehlen, diese Funktion nicht in CloudWatch Alarmen zu verwenden , wenn Sie möchten, dass die Funktion einen Skalar zurückgibt. z. B. `AVG(m2)`. Immer wenn ein Alarm auswertet , ob der Status geändert	SUM([m1,m2])/AVG(m2) AVG(METRICS())	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			werden soll, CloudWatch versucht er, eine höhere Anzahl von Datenpunk ten abzurufen als die als Bewertung szeiträume angegeben e Zahl. Diese Funktion wirkt anders, wenn zusätzliche Daten angeforde rt werden. Um diese Funktion mit Alarmen zu verwenden, insbesondere mit Alarmen, die über Auto-Scaling-Aktio nen verfügen, empfehlen wir Ihnen, den Alarm so einzustellen, dass er M von N Datenpunk		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			ten verwendet, wobei M < N ist.		
CEIL	TS TS[]	TS TS[]	Gibt die Obergrenze jeder Metrik zurück. Die Obergrenze ist die kleinste Ganzzahl, die größer oder gleich jedem Wert ist.	CEIL(m1) CEIL(METRICS()) SUM(CEIL(METRICS()))	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
DATAPOINT _COUNT	TS TS[]	S TS	Gibt die Anzahl der Datenpunkte zurück, die Werte gemeldet haben. Dies ist nützlich für die Berechnung von Durchschnittswerten für spärlichere Metriken. **Note** Wir empfehlen, diese Funktion nicht für CloudWatc h Alarme zu verwenden. Immer wenn ein Alarm auswertet , ob der Status geändert werden soll, CloudWatch versucht er, eine höhere Anzahl von Datenpunk ten abzurufen als die als Bewertung szeiträume angegebene Anzahl. Diese Funktion wirkt	SUM(m1) / DATAPOINT _COUNT(m1) DATAPOINT _COUNT(METRICS())	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			anders, wenn zusätzliche Daten angeforde rt werden.		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
DB_PERF_I NSIGHTS	Zeichen tte, Zeichen tte, Zeichen tte String, String, String[]	TS (wenn eine einzelne Zeichen lge angegel wird) TS [] (wenn ein Array von Zeichen tten angegel wird)	Liefert Zählermetriken von Performance Insights für Datenbank en wie Amazon Relational Database Service und Amazon DocumentDB (mit MongoDB-Kompatibil ität). Diese Funktion gibt dieselbe Datenmenge zurück, die Sie erhalten können, wenn Sie die Performance-Insights-APIs direkt abfragen. Sie können diese Messwerte verwenden, um CloudWatch Alarme grafisch darzustellen und zu erstellen. **Important** Wenn Sie diese Funktion verwenden, müssen Sie die eindeutig e Datenbank ressourcen-ID der Datenbank angeben. Dies unterscheidet	DB_PERF_INSIGHTS(‘ RDS’, ‘db-ABCDE FGHIJKLMN OPQRSTUVWXY1’, ‘os.cpuUtilization .user.avg’) DB_PERF_INSIGHTS(‘ DOCDB, ‘db-ABCDEFGHIJKLMN OPQRSTUVWXY1’, [‘os.cpuUtilization.idle.av g’, ‘os.cpuUtilization .user.max’])	

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			sich von der Datenbank -ID. Um die Datenbank-Ressourcen-ID in der Amazon-RDS-Konsole zu finden, wählen Sie die DB-Instance, um ihre Details anzuzeigen. Wechseln Sie zur Registerk arte Konfigura tion. Die Ressource n-ID wird im Abschnitt Konfiguration angezeigt. DB_PERF_INSIGHTS fügt die DBLoad-Metrik auch in Intervallen unter einer Minute ein. Mit dieser Funktion abgerufene Performan ce Insights Insights-Metriken werden		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			nicht in gespeichert CloudWatch. Daher funktionieren einige CloudWatch Funktione n wie kontoüber greifende Observability, Anomalieerkennung, Metrik-Streams, Metrik-Explorer und Metric Insights nicht mit Performance Insights Insights-Metriken, die Sie mit DB_PERF_I NSIGHTS abrufen. Eine einzelne Anfrage mit der Funktion DB_PERF_INSIGHTS kann die folgende Anzahl von Datenpunk ten abrufen. • 1080 Datenpunkte für Zeiträume mit hoher Auflösung (1s, 10s, 30s) • 1440 Datenpunkte für Zeiträume mit Standardauflösung (1 m, 5 m, 1 Stunde, 1 Tag)		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			Die Funktion DB_PERF_INSIGHTS unterstützt nur die folgenden Zeitrauml ängen: • 1 Sekunde • 10 Sekunden • 30 Sekunden • 1 Minute • 5 Minuten • 1 Stunde • 1 Tag Weitere Informationen zu den Zählermetriken aus Erkenntnissen zur Amazon-RDS-Leistun g finden Sie unter Zählermetriken von Performance Insights. Weitere Informationen zu den Zählermetriken aus Erkenntnissen zur Amazon-DocumentDB-Leistung finden Sie unter Zählermetriken für Performance Insights.		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			Note Hochauflösende Metriken mit Subminuten-Granularität, die von DB_PERF_INSIGHTS abgerufen werden, gelten nur für die DBLoad-Metrik oder für Betriebssystem-Metriken, wenn Sie Enhanced Monitoring mit einer höheren Auflösung aktiviert haben. Weitere Informationen zur erweiterten Überwachung von Amazon RDS finden Sie unter Überwachen von Betriebssystemmetriken mit Enhanced Monitoring.		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			Mit der Funktion DB_PERF_I NSIGHTS können Sie einen Alarm mit hoher Auflösung für einen Zeitraum von maximal drei Stunden erstellen. Sie können die CloudWatc h Konsole verwenden, um mit der Funktion DB_PERF_I NSIGHTS abgerufene Metriken für einen beliebige n Zeitraum grafisch darzustellen.		
DIFF	TS TS[]	TS TS[]	Gibt die Differenz zwischen jedem Wert in der Zeitreihe und dem vorhergehenden Wert aus dieser Zeitreihe zurück.	DIFF(m1)	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
DIFF_TIME	TS TS[]	TS TS[]	Gibt die Differenz in Sekunden zwischen dem Zeitstempel jedes Wertes in der Zeitreihe und dem Zeitstempel des vorhergehenden Wertes aus dieser Zeitreihe zurück.	DIFF_TIME(METRICS())	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
FILL	TS, [S | REPEA | LINEAR TS[], [TS | S | REPEA | LINEAR	TS TS[]	Füllt die fehlenden Werte einer Zeitreihe . Es gibt mehrere Optionen für die Werte, die als Füllstoff für fehlende Werte verwendet werden sollen: • Sie können einen Wert angeben, der als Füllwert verwendet werden soll. • Sie können eine Metrik angeben, die als Füllungswert verwendet werden soll. • Sie können das Schlüsselwort REPEAT verwenden , um fehlende Werte mit dem neuesten tatsächlichen Wert der Metrik vor dem fehlenden Wert zu füllen. • Sie können das Schlüsselwort LINEAR verwenden , um die fehlenden	FILL(m1,10) FILL(METRICS(), 0) FILL(METRICS(), m1) FILL(m1, MIN(m1)) FILL(m1, REPEAT) FILL(METRICS(), LINEAR)	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			Werte mit Werten zu füllen, die eine lineare Interpolation zwischen den Werten am Anfang und am Ende der Lücke erzeugen. **Note** Wenn Sie diese Funktion in einem Alarm verwenden , kann ein Problem auftreten, wenn Ihre Metriken mit einer leichten Verzögeru ng veröffent licht werden und die letzte Minute keine Daten enthält. In diesem Fall ersetzt FILL den fehlenden Datenpunk t durch den angeforderten		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			Wert. Dies führt dazu, dass der letzte Datenpunkt für die Metrik immer der FILL-Wert ist, was dazu führen kann, dass der Alarm entweder im OK-Zustand oder im ALARM-Zustand hängen bleibt. Sie können dies umgehen, indem Sie einen M-aus-N-Alarm verwenden . Weitere Informationen finden Sie unter Auswerten eines Alarms.		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
FIRST LAST	TS[]	TS	Gibt die erste oder letzte Zeitreihe aus einem Array von Zeitreihe n zurück. Dies ist nützlich bei Verwendun g mit der SORT-Funktion. Es kann auch verwendet werden, um die hohen und niedrigen Schwellenwerte aus der Funktion ANOMALY_D ETECTION_BAND zu erhalten.	IF(FIRST(SORT(METR ICS(), AVG, DESC))>10 0, 1, 0) Betrachtet die obere Metrik aus einem Array, das nach AVG sortiert ist. Es gibt dann eine 1 oder 0 für jeden Datenpunkt zurück, je nachdem, ob dieser Datenpunktwert über 100 liegt. AST(ANOMA LY_DETECT ION_BAND(m1)) gibt die obere Grenze des Anomalievorhersage bands zurück.	✓
FLOOR	TS TS[]	TS TS[]	Gibt die Untergrenze jeder Metrik zurück. Die Untergrenze ist die größte Ganzzahl, die kleiner oder gleich jedem Wert ist.	FLOOR(m1) FLOOR(METRICS())	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
IF	IF-Ausdruc	TS	Verwenden Sie IF zusammen mit einem Vergleichsoperator, um Datenpunkte aus einer Zeitreihe herauszufiltern oder eine gemischte Zeitreihe zu erstellen , die aus mehreren sortierten Zeitreihe n besteht. Weitere Informationen finden Sie unter Verwenden von IF-Ausdrücken.	Beispiele finden Sie unter Verwenden von IF-Ausdrücken.	✓
INSIGHT_R ULE_METRIC	INSIGH ULE_MI C(ruleN e, metricN e)	TS	Verwenden Sie INSIGHT_RULE_METRI C, um Statistiken aus einer Regel in Contribut or Insights zu extrahier en. Weitere Informati onen finden Sie unter Grafisches Darstelle n von durch Regeln generierte Metriken.		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
LAMBDA	LAMBD (Lambd nctionN e [, optional s Argume *)	TS TS{}	Ruft eine Lambda-Fu nktion auf, um Metriken aus einer Datenquel le abzufragen, die dies nicht CloudWatc h ist. Weitere Informati onen finden Sie unter So übergeben Sie Argumente an Ihre Lambda-Funktion.		
LOG	TS TS[]	TS TS[]	Der LOG einer Zeitreihe gibt den natürlichen Logarithmuswert jedes Wertes in der Zeitreihe zurück.	LOG(METRICS())	✓
LOG10	TS TS[]	TS TS[]	Der LOG10 einer Zeitreihe gibt den Logarithmuswert zur Basis 10 jedes Wertes in der Zeitreihe zurück.	LOG10(m1)	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
MAX	TS TS[]	S TS	Der Maximalwert (MAX) einer einzelnen Zeitreihe gibt eine Skalarzahl zurück, die den Maximalwert aller Datenpunkte in der Metrik darstellt. Wenn Sie ein Array von Zeitreihen eingeben, erstellt die MAX-Funkt ion eine Zeitreihe und gibt sie zurück, die aus dem höchsten Wert für jeden Datenpunk t unter den Zeitreihen besteht, die als Eingabe verwendet wurden. **Note** Wir empfehlen, diese Funktion nicht in CloudWatch Alarmen zu verwenden , wenn Sie möchten, dass die Funktion einen Skalar zurückgibt.	MAX(m1)/m1 MAX(METRICS())	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			`MAX(m2)`Wenn ein Alarm beispielsweise auswertet, ob der Status geändert werden soll, CloudWatch versucht er, eine höhere Anzahl von Datenpunk ten abzurufen als die als Bewertung szeiträume angegeben e Zahl. Diese Funktion wirkt anders, wenn zusätzliche Daten angeforde rt werden.		
METRIC_CO UNT	TS[]	S	Gibt die Anzahl der Metriken im Zeitreihen-Array zurück.	m1/METRIC _COUNT(METRICS())	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
METRICS	Null Zeichen lge	TS[]	Die Funktion METRICS () gibt alle CloudWatch Metriken in der Anfrage zurück. Mathematische Ausdrücke sind nicht enthalten. Sie können METRICS() innerhalb eines größeren Ausdrucks verwenden, der letztendl ich eine einzige Zeitreihe oder ein Array mit Zeitreihen generiert. Sie können die METRICS()-Funktion mit einer Zeichenfolge verwenden, um nur die grafisch dargestellten Metriken zurückzug eben, die diese Zeichenfolge in ihrem Feld Id (ID) enthalten. Beispielsweise gibt der Ausdruck SUM(METRI CS("errors")) eine Zeitreihe zurück, die die Summe der Werte aller grafisch dargestel lten Metriken ist, die in ihrem Feld Id (ID)	AVG(METRICS()) SUM(METRICS("error s"))	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			"errors", also Fehler, haben. Sie können auch SUM([METRICS(“4xx”), METRICS(“5xx”)]) verwenden, um mehrere Zeichenfolgen abzugleic hen.		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
MIN	TS TS[]	S TS	Der Mindestwert (MIN) einer einzelnen Zeitreihe gibt eine Skalarzahl zurück, die den Mindestwert aller Datenpunkte in der Metrik darstellt. Wenn Sie ein Array von Zeitreihen eingeben, erstellt die MIN-Funkt ion eine Zeitreihe und gibt sie zurück, die aus dem niedrigsten Wert für jeden Datenpunk t unter den Zeitreihen besteht, die als Eingabe verwendet wurden. Wenn Sie ein Array von Zeitreihen eingeben, erstellt die MIN-Funktion eine Zeitreihe und gibt sie zurück, die aus dem höchsten Wert für jeden Datenpunkt unter den Zeitreihen besteht, die als Eingabe verwendet wurden.	m1-MIN(m1) MIN(METRICS())	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			Note Wir empfehlen, diese Funktion nicht in CloudWatch Alarmen zu verwenden , wenn Sie möchten, dass die Funktion einen Skalar zurückgibt. `MIN(m2)`Wenn ein Alarm beispielsweise auswertet, ob der Status geändert werden soll, CloudWatch versucht er, eine höhere Anzahl von Datenpunk ten abzurufen als die als Bewertung szeiträume angegeben e Zahl. Diese Funktion wirkt anders, wenn		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			zusätzliche Daten angeforde rt werden.		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?

MINUTE
STUNDE
TAG
DATUM
MONAT
JAHR
EPOCHE
	TS	TS	Diese Funktionen nehmen den Zeitraum und den Bereich der Zeitreihe und geben eine neue nicht-spä rliche Zeitreihe zurück, bei der jeder Wert auf seinem Zeitstempel basiert.
			• MINUTE gibt eine nicht spärlich besetzte Zeitreihe mit ganzen Zahlen zwischen 0 und 59 zurück, die die UTC-Minute jedes Zeitstempels in der ursprünglichen Zeitreihe darstellen.
			• HOUR gibt eine nicht spärlich besetzte Zeitreihe mit ganzen Zahlen zwischen 0 und 23 zurück, die die UTC-Stunde jedes Zeitstempels in der ursprünglichen Zeitreihe darstellen.
			• DAY gibt eine nicht spärlich besetzte Zeitreihe mit ganzen
				MINUTE(m1)
				IF(DAY(m1)<6,m1) gibt nur Messwerte von Wochentagen von Montag bis Freitag zurück.
				IF(MONTH(m1) == 4,m1) gibt nur Metriken zurück, die im April veröffentlicht wurden.
					✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			Zahlen zwischen 1 und 7 zurück, die den UTC-Wochentag jedes Zeitstempels in der ursprünglichen Zeitreihe darstellen. 1 steht für Montag und 7 für Sonntag. • DATE gibt eine nicht spärlich besetzte Zeitreihe mit ganzen Zahlen zwischen 1 und 31 zurück, die den UTC-Tag des Monats jedes Zeitstempels in der ursprünglichen Zeitreihe darstellen. • MONTH gibt eine nicht spärlich besetzte Zeitreihe mit ganzen Zahlen zwischen 1 und 12 zurück, die den UTC-Monat jedes Zeitstempels in der ursprünglichen Zeitreihe darstellen. 1 steht für Januar und 12 für Dezember. • YEAR gibt eine nicht spärlich		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			besetzte Zeitreihe von Ganzzahlen zurück, die das UTC-Jahr jedes Zeitstempels in der ursprünglichen Zeitreihe darstellen. • EPOCH gibt eine nicht spärlich besetzte Zeitreihe von ganzen Zahlen zurück, die die UTC-Zeit in Sekunden seit der Epoche jedes Zeitstempels in der ursprünglichen Zeitreihe darstellt. Die Epoche ist der 1. Januar 1970.		
PERIOD	TS	S	Gibt den Zeitraum der Metrik in Sekunden zurück. Als Eingabe sind Metriken, jedoch nicht die Ergebnisse anderer Ausdrücke zulässig.	m1/PERIOD(m1)	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
RATE	TS TS[]	TS TS[]	Gibt die Änderungsrate der Metrik pro Sekunde zurück. Dieser Wert wird als Differenz zwischen dem letzten Datenpunk twert und dem vorherige n Datenpunktwert, geteilt durch die zeitliche Differenz in Sekunden zwischen den beiden Werten berechnet. **Important** Das Einstellen von Alarmen für Ausdrücke, die die RATE-Funk tion für Metriken mit spärliche n Daten verwenden , kann sich unvorhersehbar verhalten, da der Bereich der bei der Auswertung des Alarms abgerufenen Datenpunkte je nach dem	RATE(m1) RATE(METRICS())	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			Zeitpunkt der letzten Veröffent lichung der Datenpunkte variieren kann.		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
REMOVE_EMPTY	TS[]	TS[]	Entfernt alle Zeitreihen, die keine Datenpunkte haben, aus einem Array von Zeitreihen. Das Ergebnis ist ein Array von Zeitreihen, in dem jede Zeitreihe mindestens einen Datenpunkt enthält. **Note** Wir empfehlen, diese Funktion nicht für Alarme zu verwenden. CloudWatch Immer wenn ein Alarm auswertet, ob der Status geändert werden soll, CloudWatch versucht er, eine höhere Anzahl von Datenpunkten abzurufen als die als Bewertungszeiträume angegebene Anzahl. Diese	REMOVE_EMPTY(METRICS())	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			Funktion wirkt anders, wenn zusätzliche Daten angeforde rt werden.		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
RUNNING_SUM	TS TS[]	TS TS[]	Gibt eine Zeitreihe mit der laufenden Summe der Werte in der ursprünglichen Zeitreihe zurück. **Note** Wir empfehlen, diese Funktion nicht für CloudWatch Alarme zu verwenden. Immer wenn ein Alarm auswertet, ob der Status geändert werden soll, CloudWatch versucht er, eine höhere Anzahl von Datenpunkten abzurufen als die als Bewertung szeiträume angegebene Anzahl. Diese Funktion wirkt anders, wenn zusätzliche	RUNNING_SUM([m1,m2])	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			Daten angeforde rt werden.		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
SEARCH	Suchau uck	Eine oder mehrere Zeitreih n	Gibt eine oder mehrere Zeitreihen zurück, die den angegebenen Suchkriterien entsprech en. Mit der Suchfunkt ion SEARCH können Sie mehrere verwandte Zeitreihen mit nur einem Ausdruck zu einem Diagramm hinzufüge n. Das Diagramm wird dynamisch aktualisi ert, um die später hinzugefügten Metriken einzuschließen und den Suchkriterien zu entsprechen. Weitere Informationen finden Sie unter Suchausdrücke in Diagrammen verwenden . Sie können keinen Alarm basierend auf dem SEARCH-Ausdruck erstellen. Dies liegt daran, dass Suchausdr ücke mehrere Zeitreihen zurückgeben und ein auf einem mathematischen Ausdruck basierender		✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			Alarm nur eine Zeitreihe anzeigen kann. Wenn Sie bei einem Monitoring-Konto mit CloudWatch kontenübe rgreifender Observabi lity angemeldet sind, findet die SUCHFUNKT ION Metriken in den Quellkonten und im Monitoring-Konto.		
SERVICE_Q UOTA	TS, das eine Nutzung etrik ist	TS	Gibt das Serviceko ntingent für die angegebene Nutzungsm etrik zurück. Sie können dies verwenden, um Ihre aktuelle Nutzung im Vergleich zum Kontingent zu visualisi eren, und um Alarme einzustellen, die Sie warnen, wenn Sie sich dem Kontingent nähern. Weitere Informationen finden Sie unter AWS Nutzungsmetriken.		✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
SLICE	(TS[], S, S) oder (TS[], S)	TS[] TS	Ruft einen Teil eines Arrays von Zeitreihen ab. Dies ist besonders nützlich, wenn es mit SORT kombinier t wird. Beispielsweise können Sie das oberste Ergebnis aus einem Array von Zeitreihen ausschließen. Sie können zwei skalare Argumente verwenden , um den Satz von Zeitreihen zu definiere n, die zurückgegeben werden sollen. Die beiden Skalare definiere n den Anfang (inklusiv e) und Ende (exklusiv) des auszugebenden Arrays. Das Array ist null-indiziert, so dass die erste Zeitreihe im Array Zeitreihe 0 ist. Sie können auch nur einen Wert angeben und alle Zeitreihen CloudWatc h zurückgeben, die mit diesem Wert beginnen.	SLICE (SORT (METRICS (), SUM, DESC), 0, 10) gibt die 10 Metriken aus dem Array von Metriken in der Anforderung zurück, die den höchsten SUM-Wert aufweisen. SLICE (SORT (METRICS (), AVG, ASC), 5) sortiert das Array von Metriken nach der AVG-Statistik und gibt dann alle Zeitreihen mit Ausnahme der 5 mit dem niedrigsten AVG zurück.	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
SORT	(TS[], FUNCT SORT_(R) (TS[], FUNCT SORT_(R, S)	TS[]	Sortiert ein Array von Zeitreihen entsprech end der von Ihnen angegebenen Funktion. Die Funktion, die Sie verwenden, kann AVG, MIN, MAX oder SUMsein. Die Sortierre ihenfolge kann entweder ASC für aufsteigend (niedrigste Werte zuerst) oder DESC sein, um die höheren Werte zuerst anzuzeigen. Sie können optional eine Zahl nach der Sortierre ihenfolge angeben, die als Grenzwert fungiert. Wenn Sie beispiels weise einen Grenzwert von 5 angeben, werden nur die fünf wichtigsten Zeitreihen der Sortierun g zurückgegeben. Wenn diese mathemati sche Funktion in einem Diagramm angezeigt wird, werden die Beschriftungen für jede Metrik in dem Diagramm	SORT (METRICS (), AVG, DESC, 10) berechnet den Durchschnittswert jeder Zeitreihe, sortiert die Zeitreihe mit den höchsten Werten am Anfang der Sortierung und gibt nur die zehn Zeitreihen mit den höchsten Durchschn ittswerten zurück. SORT (METRICS (), MAX, ASC) sortiert das Array der Metriken nach der MAX-Statistik und gibt dann alle in aufsteigender Reihenfol ge zurück.	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			ebenfalls sortiert und nummeriert.		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
STDDEV	TS TS[]	S TS	Die Standarda bweichung (STDDEV) einer einzelnen Zeitreihe gibt eine Skalarzah l zurück, die die Standardabweichung aller Datenpunkte in der Metrik darstellt. Die Standardabweichung (STDDEV) eines Arrays von Zeitreihen gibt eine einzige Zeitreihe zurück. **Note** Wir empfehlen, diese Funktion nicht in CloudWatch Alarmen zu verwenden , wenn Sie möchten, dass die Funktion einen Skalar zurückgibt. `STDDEV(m2 )` Wenn ein Alarm beispiels weise auswertet , ob der Status geändert	m1/STDDEV(m1) STDDEV(METRICS())	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			werden soll, CloudWatch versucht er, eine höhere Anzahl von Datenpunk ten abzurufen als die als Bewertung szeiträume angegeben e Zahl. Diese Funktion wirkt anders, wenn zusätzliche Daten angeforde rt werden.		

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
SUM	TS TS[]	S TS	Die Summe (SUM) einer einzelnen Zeitreihe gibt eine Skalarzahl zurück, die die Summe der Werte aller Datenpunk te in der Metrik darstellt . Die Summe (SUM) eines Arrays von Zeitreihen gibt eine einzige Zeitreihe zurück. **Note** Wir empfehlen, diese Funktion nicht in CloudWatch Alarmen zu verwenden , wenn Sie möchten, dass die Funktion einen Skalar zurückgibt. z. B. `SUM(m1)`. Immer wenn ein Alarm auswertet , ob der Status geändert werden soll, CloudWatch versucht er, eine	SUM(METRICS())/SUM (m1) SUM([m1,m2]) SUM(METRICS("error s"))/SUM(METRICS(" requests"))*100	✓

Funktion	Argume	Typ der Rückga	Beschreibung	Beispiele	Kontoüber greifende r Support?
			höhere Anzahl von Datenpunk ten abzurufen als die als Bewertung szeiträume angegeben e Zahl. Diese Funktion wirkt anders, wenn zusätzliche Daten angeforde rt werden.		
TIME_SERIES	S	TS	Gibt eine nicht-spä rliche Zeitreihe zurück, bei der jeder Wert auf ein skalares Argument gesetzt wird.	TIME_SERIES(MAX(m1)) TIME_SERIES(5*AVG(m1)) TIME_SERIES(10)	✓

*Das Verwenden einer Funktion, die nur eine skalare Zahl zurückgibt, ist nicht gültig, da alle finalen Ergebnisse von Ausdrücken eine einzelne Zeitreihe oder ein Array von Zeitreihen sein müssen. Verwenden Sie stattdessen diese Funktionen als Teil eines größeren Ausdrucks, der eine Zeitreihe zurückgibt.

Verwenden von IF-Ausdrücken

Verwenden Sie IF zusammen mit einem Vergleichsoperator, um Datenpunkte aus einer Zeitreihe herauszufiltern oder eine gemischte Zeitreihe zu erstellen, die aus mehreren sortierten Zeitreihen besteht.

IF verwendet die folgenden Argumente:

```
IF(condition, trueValue, falseValue)
```

Die Bedingung ergibt FALSE, wenn der Wert des Bedingungsdatenpunkts 0 ist, und TRUE, wenn der Wert der Bedingung ein anderer Wert ist, unabhängig davon, ob dieser Wert positiv oder negativ ist. Wenn es sich bei der Bedingung um eine Zeitreihe handelt, wird sie für jeden Zeitstempel separat ausgewertet.

Im Folgenden werden die gültigen Syntaxregeln aufgeführt. Für jede dieser Syntaxregeln ist die Ausgabe eine einzelne Zeitreihe.

- IF(TS ***Vergleichsoperator*** S, S | TS, S | TS)

  > **Note**
  >
  > Wenn der den `TS comparison operator S` Wert TRUE hat, aber `metric2` keinen entsprechenden Datenpunkt hat, ist die Ausgabe 0.

- IF(TS, TS, TS)
- IF(TS, S, TS)
- IF(TS, TS, S)
- IF(TS, S, S)
- IF(S, TS, TS)

In den folgenden Abschnitten finden Sie weitere Details und Beispiele für diese Syntaxregeln.

IF(TS ***Vergleichsoperator*** S, scalar2 | metric2, scalar3 | metric3)

Der entsprechende Ausgabezeitreihenwert:

- hat den Wert von scalar2 oder metric2, wenn TS ***Comparison Operator*** (Vergleichsoperator) S TRUE ist
- hat den Wert von scalar3 oder metric3, wenn TS ***Comparison Operator*** (Vergleichsoperator) S FALSE ist
- hat den Wert 0, wenn der ***TS-Vergleichsoperator*** TRUE ist und der entsprechende Datenpunkt in metric2 nicht existiert.

- hat den Wert 0, wenn der ***TS-Vergleichsoperator*** FALSE ist und der entsprechende Datenpunkt in metric3 nicht existiert.
- ist eine leere Zeitreihe, wenn der entsprechende Datenpunkt von nicht in metric3, vorhanden ist oder wenn scalar3/metric3 im Ausdruck weggelassen wird

IF(metric1, metric2, metric3)

Für jeden Datenpunkt von metric1, der entsprechende Ausgabezeitreihenwert:

- hat den Wert von metric2, wenn der entsprechende Datenpunkt von metric1 TRUE ist.
- hat den Wert metric3, wenn der entsprechende Datenpunkt von metric1 FALSE ist.
- hat den Wert 0, wenn der entsprechende Datenpunkt von metric1 TRUE ist und der entsprechende Datenpunkt in metric2 nicht vorhanden ist.
- wird gelöscht, wenn der entsprechende Datenpunkt von metric1 FALSE ist und der entsprechende Datenpunkt in metric3 nicht existiert
- wird gelöscht, wenn der entsprechende Datenpunkt von metric1 FALSE ist und metric3 im Ausdruck weggelassen wird.
- wird gelöscht, wenn der entsprechende Datenpunkt von metric1 fehlt.

Die folgende Tabelle zeigt ein Beispiel für diese Syntax.

Metrik oder Funktion	Werte
(metric1)	[1, 1, 0, 0, -]
(metric2)	[30, -, 0, 0, 30]
(metric3)	[0, 0, 20, -, 20]
IF(metric1, metric2, metric3)	[30, 0, 20, 0, -]

IF(metric1, scalar2, metric3)

Für jeden Datenpunkt von metric1, der entsprechende Ausgabezeitreihenwert:

- hat den Wert von scalar2, wenn der entsprechende Datenpunkt von metric1 TRUE ist.

- hat den Wert metric3, wenn der entsprechende Datenpunkt von metric1 FALSE ist.
- wird gelöscht, wenn der entsprechende Datenpunkt von metric1 FALSE ist und der entsprechende Datenpunkt auf metric3, nicht vorhanden ist oder wenn metric3 im Ausdruck weggelassen wird.

Metrik oder Funktion	Werte
(metric1)	[1, 1, 0, 0, -]
scalar2	5
(metric3)	[0, 0, 20, -, 20]
IF(metric1, scalar2, metric3)	[5, 5, 20, -, -]

IF(metric1, metric2, scalar3)

Für jeden Datenpunkt von metric1, der entsprechende Ausgabezeitreihenwert:

- hat den Wert von metric2, wenn der entsprechende Datenpunkt von metric1 TRUE ist.
- hat den Wert von scalar3, wenn der entsprechende Datenpunkt von metric1 FALSE ist.
- hat den Wert 0, wenn der entsprechende Datenpunkt von metric1 TRUE ist und der entsprechende Datenpunkt in metric2 nicht vorhanden ist.
- wird gelöscht, wenn der entsprechende Datenpunkt in metric1 nicht vorhanden ist.

Metrik oder Funktion	Werte
(metric1)	[1, 1, 0, 0, -]
(metric2)	[30, -, 0, 0, 30]
scalar3	5
IF(metric1, metric2, scalar3)	[30, 0, 5, 5, -]

IF(scalar1, metric2, metric3)

Der entsprechende Ausgabezeitreihenwert:

- hat den Wert von metric2, wenn scalar1 TRUE ist.
- hat den Wert metric3, wenn scalar1 FALSE ist.
- ist eine leere Zeitreihe, wenn metric3 im Ausdruck weggelassen wird.

Anwendungsbeispiele für IF-Ausdrücke

Die folgenden Beispiele veranschaulichen die mögliche Verwendung der IF-Funktion .

- So zeigen Sie nur die niedrigen Werte einer Metrik an:

  IF(metric1<400, metric1)
- So ändern Sie jeden Datenpunkt in einer Metrik in einen von zwei Werten, um relative Höhen und Tiefen der ursprünglichen Metrik anzuzeigen:

  IF(metric1<400,10,2)
- So zeigen Sie eine 1 für jeden Zeitstempel an, bei dem die Latenz über dem Schwellenwert liegt, und eine 0 für alle anderen Datenpunkte:

  IF(latency>threshold, 1, 0)

Verwenden Sie metrische Mathematik bei der GetMetricData API-Operation

Sie können `GetMetricData` verwenden, um Berechnungen mithilfe mathematischer Ausdrücke durchzuführen und um große Mengen von Metrikdaten in einem einzigen API-Aufruf abzurufen. Weitere Informationen finden Sie unter GetMetricData.

Anomalieerkennung bei Metrikberechnungen

Die Anomalieerkennung bei Metrikberechnungen ist ein Feature, mit dem Sie Alarme für die Anomalieerkennung basierend auf einzelnen Metriken und metrischen mathematischen Ausdrücken erstellen können. Sie können diese Ausdrücke verwenden, um Diagramme zu erstellen, die Anomalieerkennungsbänder visualisieren. Das Feature unterstützt grundlegende arithmetische Funktionen, Vergleichs- und logische Operatoren sowie die meisten anderen Funktionen.

Die Anomalieerkennung bei Metrikberechnungen unterstützt die folgenden Funktionen nicht:

- Ausdrücke, die mehr als ein ANOMALY_DETECTION_BAND in derselben Zeile enthalten.

- Ausdrücke, die mehr als 10 Metriken oder mathematische Ausdrücke enthalten.
- Ausdrücke, die den METRICS-Ausdruck enthalten.
- Ausdrücke, die die SEARCH-Funktion enthalten.
- Ausdrücke, die die Funktion DP_PERF_INSIGHTS verwenden.
- Ausdrücke, die Metriken mit verschiedenen Perioden verwenden.
- Detektoren für metrische mathematische Anomalien, die hochauflösende Messwerte als Eingabe verwenden.

Weitere Informationen zu dieser Funktion finden Sie unter Verwenden der CloudWatch Anomalieerkennung im CloudWatch Amazon-Benutzerhandbuch.

Suchausdrücke in Diagrammen verwenden

Suchausdrücke sind mathematische Ausdrücke, die Sie zu CloudWatch Diagrammen hinzufügen können. Mit Suchausdrücken können Sie schnell mehrere verwandte Metriken zu einem Diagramm hinzufügen. Außerdem können Sie damit dynamische Diagramme erstellen, die automatisch entsprechende Metriken nutzen, die beim Erstellen des Diagramms noch nicht existieren.

Sie können beispielsweise einen Suchausdruck erstellen, der die Metrik `AWS/EC2 CPUUtilization` für alle Instances in der Region anzeigt. Wenn Sie zu einem späteren Zeitpunkt eine neue Instance starten, wird die Metrik `CPUUtilization` der neuen Instance automatisch in das Diagramm aufgenommen.

Wenn Sie einen Suchausdruck in einem Diagramm verwenden, wird dieser Ausdruck in Metriknamen, Namespaces, Dimensionsnamen und Dimensionswerten gesucht. Sie können für komplexere und leistungsstarke Suchen boolesche Operatoren verwenden. Ein Suchausdruck kann nur Metriken finden, die innerhalb der letzten zwei Wochen Daten gemeldet haben.

Sie können keinen Alarm basierend auf dem SEARCH-Ausdruck erstellen. Dies liegt daran, dass Suchausdrücke mehrere Zeitreihen zurückgeben und ein Alarm, der auf einem mathematischen Ausdruck basiert, nur eine Zeitreihe beobachten kann.

Wenn Sie ein Überwachungskonto für die CloudWatch kontenübergreifende Observability verwenden, können Sie mit Ihren Suchausdrücken Metriken in den Quellkonten finden, die mit diesem Überwachungskonto verknüpft sind.

Themen

CloudWatch Syntax der Suchausdrücke

Ein gültiger Suchausdruck hat das folgende Format.

```
SEARCH(' {Namespace, DimensionName1, DimensionName2, ...} SearchTerm', 'Statistic')
```

Zum Beispiel:

```
SEARCH('{AWS/EC2,InstanceId} MetricName="CPUUtilization"', 'Average')
```

- Der erste Teil der Abfrage in geschweiften Klammern nach dem Wort `SEARCH` ist das zu durchsuchende Metrikschema. Das Metrikschema enthält einen Metrik-Namespace und einen oder mehrere Dimensionsnamen. Die Angabe eines Metrikschemas in einer Suchabfrage ist optional. Wenn das Metrikschema angegeben wird, muss es einen Namespace enthalten und optional einen oder mehrere Dimensionsnamen, die in diesem Namespace gültig sind.

  Sie müssen keine Anführungszeichen innerhalb des Metrikschemas verwenden, es sei denn, Namespace oder Dimensionsname enthalten Leerzeichen oder nicht-alphanumerischen Zeichen. In diesem Fall müssen Sie Namen, die diese Zeichen enthalten, in doppelte Anführungszeichen einschließen.
- Auch der `SearchTerm` (Suchbegriff) ist optional. Allerdings muss eine gültige Suche entweder das Metrikschema, den `SearchTerm` oder beides enthalten. Der `SearchTerm` enthält in der Regel einen oder mehrere Konto-IDs, Metriknamen oder Dimensionswerte. Der `SearchTerm` kann mehrere Suchbegriffe enthalten; sowohl für die teilweise als auch die exakte Übereinstimmung. Er kann auch boolesche Operatoren enthalten.

  Die Verwendung einer Konto-ID in einem `SearchTerm` funktioniert nur bei Konten, die aus Gründen der CloudWatch kontenübergreifenden Beobachtbarkeit als Überwachungskonten eingerichtet sind. Die Syntax für eine Konto-ID in `SearchTerm` lautet `:aws.AccountId = "`*`444455556666`*`"`. Sie können auch `'LOCAL'` verwenden, um das Überwachungskonto selbst anzugeben: `:aws.AccountId = 'LOCAL'`

  Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.

Der `SearchTerm` kann eine oder mehrere Regionsangaben enthalten, z. B. `MetricName=` wie in diesem Beispiel; diese Angabe ist jedoch nicht erforderlich.

Das Metrikschema und der `SearchTerm` müssen gemeinsam in einfache Anführungszeichen gesetzt werden.

- Das `Statistic` ist der Name jeder gültigen CloudWatch Statistik. Er muss in einfache Anführungszeichen eingeschlossen werden. Weitere Informationen finden Sie unter Statistiken.

Das obige Beispiel sucht im Namespace `AWS/EC2` nach Metriken, die den Dimensionsnamen `InstanceId` aufweisen. Alle gefundenen `CPUUtilization`-Metriken werden zurückgegeben, wobei das Diagramm die `Average`-Statistik anzeigt.

Ein Suchausdruck kann nur Metriken finden, die innerhalb der letzten zwei Wochen Daten gemeldet haben.

Einschränkungen für Suchausdrücke

Die maximale Länge für Suchausdruckabfragen beträgt 1024 Zeichen. Sie können bis zu 100 Suchausdrücke in einem Diagramm verwenden. Ein Diagramm kann bis zu 500 Zeitreihen darstellen.

CloudWatch Suchausdrücke: Tokenisierung

Wenn Sie a angeben`SearchTerm`, sucht die Suchfunktion nach Tokens. Dabei handelt es sich um Teilzeichenfolgen, die CloudWatch automatisch aus vollständigen Metriknamen, Dimensionsnamen, Dimensionswerten und Namespaces generiert werden. CloudWatch generiert Token, die sich durch die Groß-/Kleinschreibung in der ursprünglichen Zeichenfolge unterscheiden. Auch Ziffern werden für den Beginn neuer Token verwendet. Nicht alphanumerische Zeichen dienen als Trennzeichen. Die Token werden vor und nach dem nicht alphanumerischen Zeichen erzeugt.

Eine kontinuierliche Zeichenfolge desselben Typs von Token-Trennzeichen führt zu einem Token.

Alle generierten Token bestehen aus Kleinbuchstaben. Die folgende Tabelle zeigt einige Beispiele für generierte Token.

Ursprüngliche Zeichenfolge	Erzeugte Token
CustomCount1	`customcount1` , `custom`, `count`, `1`

Ursprüngliche Zeichenfolge	Erzeugte Token
SDBFailure	`sdbfailure` , `sdb`, `failure`
Project2-trial333	`project2trial333` , `project`, `2`, `trial`, `333`

CloudWatch Suchausdrücke: Teilweise Treffer

Wenn Sie a angeben`SearchTerm`, wird der Suchbegriff ebenfalls in Tokens umgewandelt. CloudWatchfindet Metriken, die auf partiellen Übereinstimmungen basieren. Dabei handelt es sich um Übereinstimmungen eines einzelnen Tokens, das aus dem Suchbegriff generiert wurde, mit einem einzelnen Token, das aus einem Metriknamen, einem Namespace, einem Dimensionsnamen oder einem Dimensionswert generiert wurde.

Die Suche nach teilweisen Übereinstimmungen mit einem einzelnen Token unterscheidet nicht zwischen Groß- und Kleinschreibung. So finden die folgenden Suchbegriffe die Metrik `CustomCount1`:

- `count`
- `Count`
- `COUNT`

Dagegen findet der Suchbegriff `couNT` nicht die gewünschte Entsprechung `CustomCount1`, da die Groß-/Kleinschreibung im Suchbegriff `couNT` in die Token `cou` und `NT` aufgegliedert wird.

Suchen können auch zusammengesetzte Token finden, also mehrere Token, die im ursprünglichen Namen hintereinander auftreten. Die Suche nach zusammengesetzten Token unterscheidet zwischen Groß- und Kleinschreibung. Beispiel: Wenn der ursprüngliche Begriff `CustomCount1` lautet, wird dieser bei der Suche nach `CustomCount` oder `Count1` gefunden. `customcount` oder `count1` finden den Begriff dagegen nicht.

CloudWatch Suchausdrücke: Exakte Treffer

Sie können nur nach exakten Übereinstimmungen des Suchbegriffs suchen, indem Sie den Teil des Begriffs, der exakt gefunden werden soll, in doppelte Anführungszeichen setzen. Diese doppelten Anführungszeichen stehen innerhalb der einfachen Anführungszeichen, die den gesamten Suchbegriff umschließen. Zum Beispiel findet **`SEARCH(' {MyNamespace}, "CustomCount1"`**

', 'Maximum') die genaue Zeichenfolge `CustomCount1`, sofern diese als Metrikname, Dimensionsname oder Dimensionswert im Namespace `MyNamespace` vorliegt. Dagegen wird die Zeichenfolge mit **SEARCH(' {MyNamespace}, "customcount1" ', 'Maximum')** oder **SEARCH(' {MyNamespace}, "Custom" ', 'Maximum')** nicht gefunden.

Sie können teilweise und exakte Übereinstimmung in einem Suchausdruck kombinieren. Beispiel: **SEARCH(' {AWS/NetworkELB, LoadBalancer} "ConsumedLCUs" OR flow ', 'Maximum')** gibt die Elastic-Load-Balancing-Metrik mit dem Namen `ConsumedLCUs` sowie alle Elastic-Load-Balancing-Metriken oder -Dimensionen, die das Token `flow` enthalten, zurück.

Die Suche nach exakten Übereinstimmungen ist eine gute Möglichkeit zur Suche von Namen mit Sonderzeichen (z. B. nicht-alphanumerische Zeichen oder Leerzeichen), wie im folgenden Beispiel gezeigt.

```
SEARCH(' {"My Namespace", "Dimension@Name"}, "Custom:Name[Special_Characters" ',
 'Maximum')
```

CloudWatch Suchausdrücke: Ausgenommen ein Metrikschema

Alle bisherigen Beispiele enthalten ein Metrikschema in geschweiften Klammern. Aber auch Suchen, die kein Metrikschema enthalten, sind gültig.

Beispiel: **SEARCH(' "CPUUtilization" ', 'Average')** gibt alle Metriknamen, Dimensionsnamen, Dimensionswerte und Namespaces zurück, die eine exakte Übereinstimmung mit der Zeichenfolge `CPUUtilization` aufweisen. In den AWS Metrik-Namespaces kann dies Metriken von verschiedenen Diensten umfassen, darunter Amazon EC2, Amazon ECS und SageMaker andere.

Um diese Suche auf nur einen AWS Service zu beschränken, empfiehlt es sich, den Namespace und alle erforderlichen Dimensionen im Metrikschema anzugeben, wie im folgenden Beispiel. Damit wird die Suche zwar auf den Namespace `AWS/EC2` begrenzt, aber es werden weiterhin Ergebnisse für andere Metriken ausgegeben, wenn Sie `CPUUtilization` als Dimensionswert für diese Metriken definiert haben.

```
SEARCH(' {AWS/EC2, InstanceType} "CPUUtilization" ', 'Average')
```

Alternativ können Sie den Namespace im `SearchTerm` übergeben, wie im folgenden Beispiel gezeigt. Allerdings gibt diese Suche alle Zeichenfolgen mit `AWS/EC2` zurück, auch wenn es sich um einen benutzerdefinierten Dimensionsnamen oder -wert handelt.

```
SEARCH(' "AWS/EC2" MetricName="CPUUtilization" ', 'Average')
```

CloudWatch Suchausdrücke: Angabe von Eigenschaftsnamen bei der Suche

Die folgende Suche nach einer exakten Übereinstimmung mit `"CustomCount1"` gibt alle Metriken mit genau diesem Namen zurück.

```
SEARCH(' "CustomCount1" ', 'Maximum')
```

Aber sie gibt auch Metriken mit Dimensionsnamen, Dimensionswerten oder Namespaces zurück, die `CustomCount1` enthalten. Zur weiteren Strukturierung der Suche können Sie den Eigenschaftenname für den gesuchten Objekttyp in die Suche einbeziehen. Im folgenden Beispiel werden alle Namespaces nach Metriken mit dem Namen `CustomCount1` durchsucht.

```
SEARCH(' MetricName="CustomCount1" ', 'Maximum')
```

Sie können auch Namespaces und Paare mit Dimensionsname/Wert als Eigenschaftsnamen verwenden, wie in den folgenden Beispielen dargestellt. Das erste dieser Beispiele zeigt auch, dass Sie nach Eigenschaftennamen mit teilweisen Übereinstimmungen suchen können.

```
SEARCH(' InstanceType=micro ', 'Average')
```

```
SEARCH(' InstanceType="t2.micro" Namespace="AWS/EC2" ', 'Average')
```

CloudWatch Suchausdrücke: Nicht-alphanumerische Zeichen

Nicht-alphanumerische Zeichen dienen als Trennzeichen und markieren so, wo die Namen von Metriken, Dimensionen, Namespaces und Suchbegriffen in Token aufgegliedert werden. Beim Aufgliedern von Begriffen werden nicht alphanumerische entfernt und nicht in Token übernommen. Beispiel: `Network-Errors_2` generiert die Token `network`, `errors` und `2`.

Ihr Suchbegriff kann beliebige nicht alphanumerische Zeichen enthalten. Wenn diese Zeichen in Ihrem Suchbegriff enthalten sind, können sie zusammengesetzte Token in teilweisen Übereinstimmungen angeben. Beispiel: Die folgenden Suchen finden Metriken mit dem Namen `Network-Errors-2` oder `NetworkErrors2`.

```
network/errors
network+errors
```

```
network-errors
Network_Errors
```

Wenn Sie einen exakten Wert suchen, müssen alle nicht alphanumerischen Zeichen in der exakten Suche mit den in der gesuchten Zeichenfolge verwendeten Zeichen übereinstimmen. Beispiel: Wenn Sie nach `Network-Errors-2` suchen, führt die Suche nach `"Network-Errors-2"` zum Erfolg, nicht aber die Suche nach `"Network_Errors_2"`.

Wenn Sie eine exakte Übereinstimmung suchen, müssen die folgenden Zeichen durch den umgekehrten Schrägstrich als Escape-Zeichen markiert werden.

```
" \ ( )
```

Um beispielsweise den Metriknamen `Europe\France Traffic(Network)` als exakte Übereinstimmung zu finden, muss der Suchbegriff **`"Europe\\France Traffic\(Network\)"`** lauten.

CloudWatch Suchausdrücke: Boolesche Operatoren

Sie können in der Suche die booleschen Operatoren `AND`, `OR` und `NOT` im `SearchTerm` verwenden. Boolesche Operatoren werden innerhalb der einfachen Anführungszeichen, die den gesamten Suchbegriff einschließen, verwendet. Boolesche Operatoren unterscheiden zwischen Groß- und Kleinschreibung. `and`, `or` und `not` sind somit keine gültigen booleschen Operatoren.

Sie können `AND` explizit in der Suche verwenden, z. B. **`SEARCH('{AWS/EC2,InstanceId} network AND packets', 'Average')`**. Wenn Sie keinen booleschen Operator zwischen Suchbegriffen angeben, wird implizit der Operator `AND` verwendet. **`SEARCH(' {AWS/EC2,InstanceId} network packets ', 'Average')`** führt also zum selben Ergebnis.

Mit `NOT` können Sie Datenteilmengen aus den Ergebnissen ausschließen. Beispiel: **`SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" NOT i-1234567890123456 ', 'Average')`** gibt die `CPUUtilization` für alle Instances mit Ausnahme der Instance `i-1234567890123456` aus. Sie können auch einen `NOT`-Ausdruck als einzigen Suchbegriff verwenden. Beispiel: **`SEARCH( 'NOT Namespace=AWS ', 'Maximum')`** gibt alle benutzerdefinierten Metriken aus (Metriken mit Namespaces, in denen `AWS` nicht vorkommt).

Sie können mehrere `NOT`-Ausdrücke in einer Abfrage verwenden. Beispiel: **`SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" NOT "ProjectA" NOT "ProjectB" ', 'Average')`** gibt die `CPUUtilization` aller Instances in der Region mit Ausnahme der Instances aus, die einen Dimensionswert von `ProjectA` oder `ProjectB` enthalten.

Sie können die booleschen Operatoren kombinieren, um weitere leistungsstarke und detaillierte Suchen zu erstellen (vgl. die folgenden Beispiele). Verwenden Sie Klammern zum Gruppieren der Operatoren.

Die beiden nächsten Beispiele geben alle Metriknamen mit der Zeichenfolge `ReadOps` aus den Namespaces EC2 und EBS aus.

```
SEARCH(' (EC2 OR EBS) AND MetricName=ReadOps ', 'Maximum')
```

```
SEARCH(' (EC2 OR EBS) MetricName=ReadOps ', 'Maximum')
```

Im folgenden Beispiel wird die vorherige Suche auf Ergebnisse beschränkt, die `ProjectA` als Wert einer Dimension enthalten.

```
SEARCH(' (EC2 OR EBS) AND ReadOps AND ProjectA ', 'Maximum')
```

Das folgenden Beispiel nutzt verschachtelte Gruppierungen. Es gibt Lambda-Metriken für Fehler (`Errors`) aus allen Funktionen und Funktionsaufrufe (`Invocations`) mit Namen, die die Zeichenfolgen `ProjectA` oder `ProjectB` enthalten, aus.

```
SEARCH(' {AWS/Lambda,FunctionName} MetricName="Errors" OR (MetricName="Invocations" AND
 (ProjectA OR ProjectB)) ', 'Average')
```

CloudWatch Suchausdrücke: Verwendung mathematischer Ausdrücke

Sie können einen Suchausdruck als Teil von mathematischen Ausdrücken in einem Diagramm verwenden.

Beispiel: **`SUM(SEARCH(' {AWS/Lambda, FunctionName} MetricName="Errors" ', 'Sum'))`** gibt die Summe der `Errors`-Metrik für alle Ihre Lambda-Funktionen zurück.

Mit separaten Zeilen für Such- und mathematische Ausdrücke erzielen Sie eventuell nützlichere Ergebnisse. Angenommen, Sie verwenden die beiden folgenden Ausdrücke in einem Diagramm. Die erste Zeile zeigt separate `Errors`-Linien für jede Ihrer Lambda-Funktionen an. Die ID dieses Ausdrucks ist `e1`. Die zweite Zeile fügt eine weitere Linie mit den Summen der Fehler aller Funktionen hinzu.

```
SEARCH(' {AWS/Lambda, FunctionName}, MetricName="Errors" ', 'Sum')
SUM(e1)
```

CloudWatch Beispiele für Suchausdrücke

Die folgenden Beispiele veranschaulichen weitere Aspekte für Suchausdrücke und Syntax. Den Anfang macht ein Beispiel für die Suche nach `CPUUtilization` in allen Instances der Region. Daran schließen sich Varianten dieser Suche an.

Dieses Beispiel zeigt für jede Instance in der Region eine Linie für die Metrik `CPUUtilization` aus dem Namespace `AWS/EC2` an.

```
SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" ', 'Average')
```

Durch Ändern von `InstanceType` in `InstanceId` enthält das Diagramm eine Linie für jeden in der Region verwendeten Instance-Typ. Daten aus allen Instances pro Typ werden zu einer Linie für diesen Instance-Typ aggregiert.

```
SEARCH(' {AWS/EC2,InstanceType} MetricName="CPUUtilization" ', 'Average')
```

Das folgende Beispiel aggregiert die `CPUUtilization` nach Instance-Typ und zeigt eine Linie für jeden Instance-Typ, der die Zeichenfolge `micro` enthält, an.

```
SEARCH('{AWS/EC2,InstanceType} InstanceType=micro MetricName="CPUUtilization" ',
 'Average')
```

Dieses Beispiel schränkt das vorherige Beispiel ein, indem `InstanceType` auf exakte Übereinstimmung mit „t2.micro“ geprüft wird.

```
SEARCH('{AWS/EC2,InstanceType} InstanceType="t2.micro" MetricName="CPUUtilization" ',
 'Average')
```

Die folgende Suche entfernt den Teil `{metric schema}` der Abfrage, sodass die Metrik `CPUUtilization` aus allen Namespaces im Diagramm angezeigt wird. Dies kann zu einer ganzen Reihe von Ergebnissen führen, da das Diagramm mehrere Zeilen für die `CPUUtilization` Metrik aus jedem AWS Service enthält, die nach verschiedenen Dimensionen aggregiert sind.

```
SEARCH('MetricName="CPUUtilization" ', 'Average')
```

Um diese Ergebnisse einzugrenzen, können Sie zwei bestimmte Metrik-Namespaces angeben.

```
SEARCH('MetricName="CPUUtilization" AND ("AWS/ECS" OR "AWS/ES") ', 'Average')
```

Das vorherige Beispiel ist die einzige Möglichkeit, mehrere Namespaces mit einer Suchabfrage zu durchsuchen, da Sie nur ein Metrikschema pro Abfrage verwenden können. Um die Abfrage besser zu strukturieren, könnten Sie jedoch zwei Abfragen im Diagramm verwenden, wie im folgenden Beispiel gezeigt. Dieses Beispiel sorgt zudem durch Angabe einer Dimension für die Aggregation der Daten aus Amazon ECS für mehr Struktur.

```
SEARCH('{AWS/ECS ClusterName}, MetricName="CPUUtilization" ', 'Average')
SEARCH(' {AWS/EBS} MetricName="CPUUtilization" ', 'Average')
```

Das folgende Beispiel gibt die Elastic-Load-Balancing-Metrik mit dem Namen `ConsumedLCUs` sowie alle Elastic-Load-Balancing-Metriken oder -Dimensionen zurück, die das Token `flow` enthalten.

```
SEARCH('{AWS/NetworkELB, LoadBalancer} "ConsumedLCUs" OR flow ', 'Maximum')
```

Das folgenden Beispiel nutzt verschachtelte Gruppierungen. Es gibt Lambda-Metriken für Fehler (`Errors`) aus allen Funktionen und Funktionsaufrufe (`Invocations`) mit Namen, die die Zeichenfolgen `ProjectA` oder `ProjectB` enthalten, aus.

```
SEARCH('{AWS/Lambda,FunctionName} MetricName="Errors" OR (MetricName="Invocations" AND
 (ProjectA OR ProjectB)) ', 'Average')
```

Das folgende Beispiel zeigt alle benutzerdefinierten Metriken an, ausgenommen von AWS -Services erzeugte Metriken.

```
SEARCH('NOT Namespace=AWS ', 'Average')
```

Das folgende Beispiel zeigt Metriken mit Metriknamen, Namespaces, Dimensionsnamen und Dimensionswerten an, die die Zeichenfolge `Errors` im Namen enthalten.

```
SEARCH('Errors', 'Average')
```

Das folgende Beispiel beschränkt die Suche auf exakte Übereinstimmungen. Diese Suche finden zum Beispiel den Metriknamen `Errors`, aber keine Metriken namens `ConnectionErrors` oder `errors`.

```
SEARCH(' "Errors" ', 'Average')
```

Das folgende Beispiel zeigt, wie Namen mit Leer- oder Sonderzeichen im Metrikschema des Suchbegriffs angegeben werden.

```
SEARCH('{"Custom-Namespace", "Dimension Name With Spaces"}, ErrorCount ', 'Maximum')
```

CloudWatch Beispiele für kontenübergreifende Observabilitäts-Suchausdrücke

CloudWatch Beispiele für kontenübergreifende Beobachtbarkeit

Wenn Sie bei einem Konto angemeldet sind, das als Überwachungskonto für CloudWatch kontenübergreifende Observability eingerichtet ist, können Sie die SUCHFUNKTION verwenden, um Metriken von bestimmten Quellkonten zurückzugeben. Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.

Das folgende Beispiel ruft alle Lambda-Metriken aus dem Konto mit der Konto-ID 111122223333 ab.

```
SEARCH(' AWS/Lambda :aws.AccountId = "111122223333" ', 'Average')
```

Das folgende Beispiel ruft alle AWS/EC2-Metriken von zwei Konten ab: 111122223333 und 777788889999.

```
SEARCH(' AWS/EC2 :aws.AccountId = ("111122223333" OR "777788889999") ', 'Average')
```

Das folgende Beispiel ruft alle AWS/EC2-Metriken aus dem Quellkonto 111122223333 und aus dem Überwachungskonto selbst ab.

```
SEARCH(' AWS/EC2 :aws.AccountId = ("111122223333" OR 'LOCAL') ', 'Average')
```

Das folgende Beispiel ruft die SUM der MetaDataToken-Metrik von dem Konto 444455556666 mit der InstanceId-Dimension ab.

```
SEARCH('{AWS/EC2,InstanceId} :aws.AccountId=444455556666 MetricName=\"MetadataNoToken
\"','Sum')
```

Erstellen Sie ein CloudWatch Diagramm mit einem Suchausdruck

In der CloudWatch Konsole können Sie auf die Suchfunktion zugreifen, wenn Sie einem Dashboard ein Diagramm hinzufügen, oder indem Sie die Metrikansicht verwenden.

Sie können keinen Alarm basierend auf dem SEARCH-Ausdruck erstellen. Dies liegt daran, dass Suchausdrücke mehrere Zeitreihen zurückgeben und ein auf einem mathematischen Ausdruck basierender Alarm nur eine Zeitreihe anzeigen kann.

So fügen Sie ein Diagramm mit einem Suchausdruck zu einem vorhandenen Dashboard hinzu

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Dashboards aus und wählen Sie dann ein Dashboard.
3. Wählen Sie Add widget aus.
4. Klicken Sie entweder auf Line (Linie) oder Stacked area (Gestapelter Bereich) und anschließend auf Configure (Konfigurieren).
5. Wählen Sie auf der Registerkarte Graphed metrics (Grafisch dargestellte Metriken) die Option Add a math expression (Mathematischen Ausdruck hinzufügen) aus.
6. Geben Sie unter Details den gewünschten Suchausdruck ein. Zum Beispiel, **`SEARCH('{AWS/EC2,InstanceId} MetricName="CPUUtilization"', 'Average')`**
7. (Optional) Um einen weiteren Such- oder mathematischen Ausdruck zum Diagramm hinzuzufügen, wählen Sie Add math expression (Mathematischen Ausdruck hinzufügen) aus.
8. (Optional) Nachdem Sie einen Suchausdruck hinzugefügt haben, können Sie für jede Metrik eine dynamische Beschriftung angeben, die in der Diagrammlegende angezeigt wird. Dynamische Beschriftungen zeigen eine Statistik über die Metrik an und werden automatisch aktualisiert, wenn das Dashboard oder das Diagramm aktualisiert wird. Um eine dynamische Beschriftung hinzuzufügen, wählen Sie Graphed metrics (Dargestellte Metriken) und dann Dynamic labels (Dynamische Beschriftungen).

   Standardmäßig werden die dynamischen Werte, die Sie der Beschriftung hinzufügen, am Anfang der Beschriftung angezeigt. Sie können dann auf den Wert Label (Beschriftung) für die Metrik klicken, um die Beschriftung zu bearbeiten. Weitere Informationen finden Sie unter Dynamische Labels verwenden.
9. (Optional) Um eine einzelne Metrik zum Diagramm hinzuzufügen, wechseln Sie zur Registerkarte All metrics (Alle Metriken) und arbeiten sich zur gewünschten Metrik vor.
10. (Optional) Um den im Diagramm dargestellten Zeitraum zu ändern, wählen Sie entweder oben im Diagramm custom (benutzerdefiniert) oder einen der Zeiträume links neben dem Eintrag custom (benutzerdefiniert) aus.
11. (Optional) Mithilfe von horizontalen Anmerkungen können Dashboard-Benutzer bei Metriken schnell Spitzenwerte auf einer bestimmten Stufe erkennen, oder ob eine Metrik in einem

vordefinierten Bereich liegt. Wenn Sie eine horizontale Anmerkung hinzufügen möchten, wählen Sie Graph options (Diagrammoptionen) und anschließend Add horizontal annotation (Horizontale Anmerkung einfügen) aus.

a. Geben Sie unter Label (Bezeichnung) eine Bezeichnung für die Anmerkung ein.

b. Geben Sie unter Value (Wert) den Metrikwert ein, an dem die horizontale Anmerkung angezeigt wird.

c. Geben Sie unter Fill an, ob die Füllschattierung bei der Anmerkung verwendet werden soll. Wählen Sie beispielsweise `Above` oder `Below` als entsprechend zu füllenden Bereich aus. Wenn Sie `Between` angeben, wird ein anderes `Value`-Feld angezeigt und der Bereich des Diagramms zwischen zwei Werten wird gefüllt.

d. Geben Sie für Axis (Achse) an, ob sich die Nummern in `Value` auf die der linken Y-Achse oder der rechten Y-Achse zugewiesenen Metrik beziehen, wenn das Diagramm mehrere Metriken enthält.

   Sie können die Füllfarbe einer Anmerkung ändern, indem Sie das Farbquadrat in der linken Spalte neben der Anmerkung auswählen.

Wiederholen Sie die Schritte, um einem Diagramm mehrere horizontale Anmerkungen hinzuzufügen.

Um eine Anmerkung auszublenden, deaktivieren Sie das Kontrollkästchen in der linken Spalte für diese Anmerkung.

Um eine Anmerkung zu löschen, wählen Sie das x in der Spalte Actions aus.

12. (Optional) Mithilfe von vertikalen Anmerkungen können Sie Meilensteine in einem Diagramm markieren, beispielsweise Betriebsereignisse oder Anfang und Ende einer Bereitstellung. Wenn Sie eine vertikale Anmerkung hinzufügen möchten, klicken Sie auf Graph options (Diagrammoptionen) und dann auf Add vertical annotation (Vertikale Anmerkung hinzufügen):

    a. Geben Sie unter Label (Bezeichnung) eine Bezeichnung für die Anmerkung ein. Wenn Sie nur das Datum und die Uhrzeit in der Anmerkung anzeigen möchten, lassen Sie das Feld Label (Beschriftung) leer.

    b. Geben Sie für Date (Datum) das Datum und die Uhrzeit an, zu dem die vertikale Anmerkung angezeigt werden soll.

    c. Geben Sie für Fill (Ausfüllen) an, ob vor oder nach einer vertikalen Anmerkung oder zwischen zwei vertikalen Anmerkungen eine Füllschattierung verwendet werden soll.

Wählen Sie beispielsweise `Before` oder `After` als entsprechend zu füllenden Bereich aus. Wenn Sie `Between` angeben, wird ein anderes `Date`-Feld angezeigt und der Bereich des Diagramms zwischen zwei Werten wird gefüllt.

Wiederholen Sie die Schritte, um einem Diagramm mehrere vertikale Anmerkungen hinzuzufügen.

Um eine Anmerkung auszublenden, deaktivieren Sie das Kontrollkästchen in der linken Spalte für diese Anmerkung.

Um eine Anmerkung zu löschen, wählen Sie das x in der Spalte Actions aus.

13. Wählen Sie Create widget aus.
14. Wählen Sie Save dashboard aus.

So stellen Sie durchsuchte Metriken in der Ansicht für Metriken grafisch dar

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Geben Sie im Suchfeld die zu suchenden Token an, z. B. **`cpuutilization t2.small`**.

   Passende Ergebnisse werden angezeigt.
4. Um alle Metriken, die den Suchkriterien entsprechen, grafisch darzustellen, wählen Sie Graph search (Graphensuche) aus.

   or

   Wählen Sie zum Verfeinern der Suche einen der Namespaces aus, die in den Suchergebnissen angezeigt wurden.
5. Bei Wahl eines Namespaces zum Eingrenzen der Ergebnisse haben Sie folgende Möglichkeiten:

   a. Um eine oder mehrere Metriken grafisch darzustellen, aktivieren Sie das Kontrollkästchen neben jeder Metrik. Um alle Metriken auszuwählen, aktivieren Sie das Kontrollkästchen in der Kopfzeile der Tabelle.

   b. Zeigen Sie zum Verfeinern der Suche mit dem Mauszeiger auf den Namen einer Metrik und wählen Sie Add to search (Zu Suche hinzufügen) oder Search for this only (Nur hiernach suchen) aus.

c. Wenn Sie Hilfe für eine Metrik anzeigen möchten, wählen Sie den Metriknamen und anschließend What is this? (Was ist das?) aus.

Die ausgewählten Metriken werden im Diagramm dargestellt.

6. (Optional) Verwenden Sie eine der Schaltflächen auf der Suchleiste, um den entsprechenden Teil des Suchbegriffs zu bearbeiten.
7. (Optional) Um das Diagramm einem Dashboard hinzuzufügen, wählen Sie Actions (Aktionen) gefolgt von Add to dashboard (Zum Dashboard hinzufügen) aus.

Abrufen von Statistiken für eine Metrik

CloudWatch Definitionen von Statistiken

Statistiken sind Metrikdaten, die über bestimmte Zeiträume aggregiert wurden. Wenn Sie die Statistiken für eine Metrik grafisch darstellen oder abrufen, geben Sie den Zeitraum an, z. B. fünf Minuten, der zum Berechnen jedes statistischen Werts verwendet werden soll. Wenn der Zeitraum beispielsweise fünf Minuten beträgt, ist die Summe die Summe aller während des Fünf-Minuten-Zeitraums erfassten Stichprobenwerte, während das Minimum der niedrigste Wert ist, der während des Fünf-Minuten-Zeitraums erfasst wurde.

CloudWatch unterstützt die folgenden Statistiken für Metriken.

- SampleCountist die Anzahl der Datenpunkte während des Zeitraums.
- Summe ist die Summe der Werte aller während des Zeitraums gesammelten Datenpunkte.
- Durchschnitt ist der Wert von `Sum/SampleCount` während des angegebenen Zeitraums.
- Minimum ist der niedrigste Wert, der während des angegebenen Zeitraums beobachtet wurde.
- Maximum ist der höchste während des angegebenen Zeitraums beobachtete Wert.
- Perzentil (p) gibt die relative Stellung eines Werts in einem Datensatz an. p95 ist beispielsweise das 95. Perzentil und bedeutet, dass 95 Prozent der Daten innerhalb des Zeitraums unter diesem Wert und 5 Prozent der Daten über diesem Wert liegen. Perzentile verhelfen Ihnen zu einem besseren Verständnis für die Verteilung Ihrer Metrikdaten.
- Getrimmter Mittelwert (TM) ist der Mittelwert aller Werte, die zwischen zwei angegebenen Grenzen liegen. Werte außerhalb der Grenzen werden ignoriert, wenn der Mittelwert berechnet wird. Sie definieren die Grenzen als eine oder zwei Zahlen zwischen 0 und 100, bis zu 10 Dezimalstellen.

Die Zahlen können absolute Werte oder Prozentsätze sein. tm90 berechnet beispielsweise den Durchschnitt, nachdem die 10 % der Datenpunkte mit den höchsten Werten entfernt wurden. TM(2%:98%) berechnet den Durchschnitt, nachdem die 2 % niedrigsten Datenpunkte und die 2 % höchsten Datenpunkte entfernt wurden. TM(150:1000) berechnet den Durchschnitt, nachdem alle Datenpunkte entfernt wurden, die kleiner oder gleich 150 oder größer als 1 000 sind.

- Interquartil-Mittelwert (IQM) ist der getrimmte Mittelwert des Interquartil-Bereichs oder die mittleren 50 % der Werte. Es entspricht TM(25%:75%).
- Der Winsorized-Mittelwert (WM) ähnelt dem getrimmten Mittelwert. Beim Winsorized-Mittelwert werden jedoch die Werte, die außerhalb der Grenze liegen, nicht ignoriert, sondern als gleich dem Wert am Edge der entsprechenden Grenze betrachtet. Nach dieser Normalisierung wird der Durchschnitt berechnet. Sie definieren die Grenzen als eine oder zwei Zahlen zwischen 0 und 100, bis zu 10 Dezimalstellen. Zum Beispiel berechnet wm98 den Durchschnitt, während die 2 % der höchsten Werte gleich dem Wert im 98. Perzentil behandelt werden. WM(10%:90%) berechnet den Durchschnitt, wobei die höchsten 10 % der Datenpunkte als Wert der 90 %-Grenze und die niedrigsten 10 % der Datenpunkte als Wert der 10 %-Grenze behandelt werden.
- Der Perzentilrang (PR) ist der Prozentsatz der Werte, die einen festen Schwellenwert erfüllen. PR(:300) gibt beispielsweise den Prozentsatz der Datenpunkte mit einem Wert von 300 oder weniger zurück. PR(100:2000) gibt den Prozentsatz der Datenpunkte mit einem Wert zwischen 100 und 2 000 zurück.

  Der Perzentilrang ist an der Untergrenze ausschließlich und an der Obergrenze inklusiv.
- Getrimmter Zähler (TC) ist die Anzahl der Datenpunkte im ausgewählten Bereich für eine getrimmte Mittelwertstatistik. tc90 gibt beispielsweise die Anzahl der Datenpunkte zurück, ohne Datenpunkte, die in die höchsten 10 % der Werte fallen. TC(0.005:0.030) gibt die Anzahl der Datenpunkte mit Werten zwischen 0,005 (exklusiv) und 0,030 (inklusive) zurück.
- Getrimmte Summe (TS) ist die Summe der Werte von Datenpunkten in einem ausgewählten Bereich für eine getrimmte Mittelwertstatistik. Er entspricht (getrimmter Mittelwert) * (getrimmter Zählwert). tc90 gibt beispielsweise die Summe der Datenpunkte zurück, ohne Datenpunkte, die in die höchsten 10 % der Werte fallen. TS(80%:) gibt die Summe der Datenpunktwerte zurück, ohne Datenpunkte mit Werten in den niedrigsten 80 % des Wertebereichs.

> **Note**
>
> Wenn Sie für getrimmter Mittelwert, getrimmte Anzahl, getrimmte Summe und Winsorized-Mittelwert zwei Grenzen als feste Werte anstelle von Prozentsätzen definieren, umfasst die

Berechnung Werte, die der oberen Grenze entsprechen, jedoch keine Werte, die der unteren Grenze entsprechen.

Syntax

Für Getrimmter Mittelwert, Getrimmte Anzahl, Getrimmte Summe und Winsorized-Mittelwert gelten die folgenden Syntaxregeln:

- Durch die Verwendung von Klammern mit einer oder zwei Zahlen mit Prozentzeichen werden die Grenzen definiert, die als Werte im Datensatz verwendet werden, die zwischen den beiden von Ihnen angegebenen Perzentilen liegen. Beispielsweise verwendet TM(10%:90%) nur die Werte zwischen dem 10. und 90. Perzentil. TM(:95%) verwendet die Werte vom untersten Ende des Datensatzes bis zum 95. Perzentil und ignoriert die 5 % der Datenpunkte mit den höchsten Werten.
- Die Verwendung von Klammern mit einer oder zwei Zahlen ohne Prozentzeichen definiert die Grenzen, die als Werte im Datensatz verwendet werden, die zwischen den expliziten Werten liegen, die Sie angeben. TC(80:500) verwendet beispielsweise nur die Werte zwischen 80 (exklusiv) und 500 (inklusive). TC(:0.5) verwendet nur die Werte, die 0,5 gleich oder niedriger sind.
- Bei Verwendung einer Zahl ohne Klammern wird mit Prozentsätzen berechnet, wobei Datenpunkte ignoriert werden, die über dem angegebenen Perzentil liegen. Beispielsweise berechnet tm99 den Mittelwert und ignoriert dabei die 1 % der Datenpunkte mit dem höchsten Wert. Es ist das gleiche wie TM(:99%).
- Getrimmter Mittelwert, Getrimmte Anzahl, Getrimmte Summe und Winsorized Mean können alle mit Großbuchstaben abgekürzt werden, wenn ein Bereich angegeben wird, z. B. TM(5%:95%), TM(100:200) oder TM(:95%). Sie können nur mit Kleinbuchstaben abgekürzt werden, wenn Sie nur eine Zahl angeben, z. B. tm99.

Anwendungsfälle für Statistiken

- Der getrimmte Mittelwert ist am nützlichsten für Metriken mit einer großen Stichprobengröße, wie z. B. die Webseiten-Latenz. Beispielsweise ignoriert tm99 extrem hohe Ausreißer, die aus Netzwerkproblemen oder menschlichen Fehlern resultieren könnten, um eine genauere Zahl für die durchschnittliche Latenzzeit typischer Anfragen zu erhalten. In ähnlicher Weise ignoriert TM(10%:) die niedrigsten 10 % der Latenzwerte, wie sie beispielsweise aus Cache-Treffern resultieren. Und TM (10%:99%) schließt diese beiden Arten von Ausreißern aus. Es wird empfohlen, für die Überwachung der Latenz den getrimmten Mittelwert zu verwenden.

- Es empfiehlt sich, bei getrimmten Mittelwerten stets auf die getrimmte Anzahl zu achten, um sicherzustellen, dass die Anzahl der Werte, die in den getrimmten Mittelwerten verwendet werden, ausreicht, um statistisch signifikant zu sein.
- Der Perzentil-Rang ermöglicht es Ihnen, Werte in „Abschnitte“ von Bereichen zu setzen, und Sie können diese verwenden, um manuell ein Histogramm zu erstellen. Teilen Sie dazu Ihre Werte in verschiedene Klassen auf, z. B. PR(:1), PR(1:5), PR(5:10) und PR(10:). Setzen Sie jede dieser Abschnitte in eine Visualisierung als Balkendiagramme, und Sie haben ein Histogramm.

  Der Perzentilrang ist an der Untergrenze ausschließlich und an der Obergrenze inklusiv.

Perzentile im Vergleich zum getrimmten Mittelwert

Ein Perzentil wie p99 und ein getrimmter Mittelwert wie tm99 messen ähnliche, aber nicht identische Werte. Sowohl p99 als auch tm99 ignorieren die 1 % der Datenpunkte mit den höchsten Werten, die als Ausreißer gelten. Danach ist p99 der Maximalwert der verbleibenden 99 %, während tm99 der Durchschnitt der verbleibenden 99 % ist. Wenn Sie sich die Latenz von Webanfragen ansehen, sagt Ihnen p99 die schlechteste Kundenerfahrung und ignoriert Ausreißer, während tm99 die durchschnittliche Kundenerfahrung anzeigt und Ausreißer ignoriert.

Der getrimmte Mittelwert ist eine gute Latenzstatistik, die Sie sich ansehen sollten, wenn Sie Ihr Kundenerlebnis optimieren möchten.

Anforderungen für die Verwendung von Perzentilen, getrimmtem Mittelwert und einigen anderen Statistiken

CloudWatch benötigt Rohdatenpunkte, um die folgenden Statistiken zu berechnen:

- Perzentile
- Getrimmter Mittelwert
- Interquartil-Mittelwert
- Winsorized-Mittelwert
- Getrimmte Summe
- Getrimmte Anzahl
- Perzentilrang

Wenn Sie Daten für eine benutzerdefinierte Statistik mit einem Statistiksatz anstelle von Rohdaten veröffentlichen, können Sie diese Statistiktypen für diese Daten nur abrufen, wenn eine der folgenden Bedingungen zutrifft:

- Der SampleCount Wert des Statistiksatzes ist 1 und Min, Max und Summe sind alle gleich.
- Min und Max sind gleich, und Summe ist gleich Min multipliziert mit. SampleCount

Die folgenden AWS Dienste umfassen Metriken, die diese Arten von Statistiken unterstützen.

- API Gateway
- Application Load Balancer
- Amazon EC2
- Elastic Load Balancing
- Kinesis
- Amazon RDS

Darüber hinaus sind diese Statistiktypen für Metriken nicht verfügbar, wenn einer der Metrikwerte negative Zahlen ist.

Die folgenden Beispiele zeigen Ihnen, wie Sie Statistiken für die CloudWatch Metriken für Ihre Ressourcen, wie z. B. Ihre EC2-Instances, abrufen können.

Beispiele

- Statistiken für eine bestimmte Ressource abrufen
- Statistiken zwischen Ressourcen aggregieren
- Aggregieren von Statistiken nach Auto Scaling-Gruppe
- Statistiken nach Amazon Machine Image (AMI) aggregieren

Statistiken für eine bestimmte Ressource abrufen

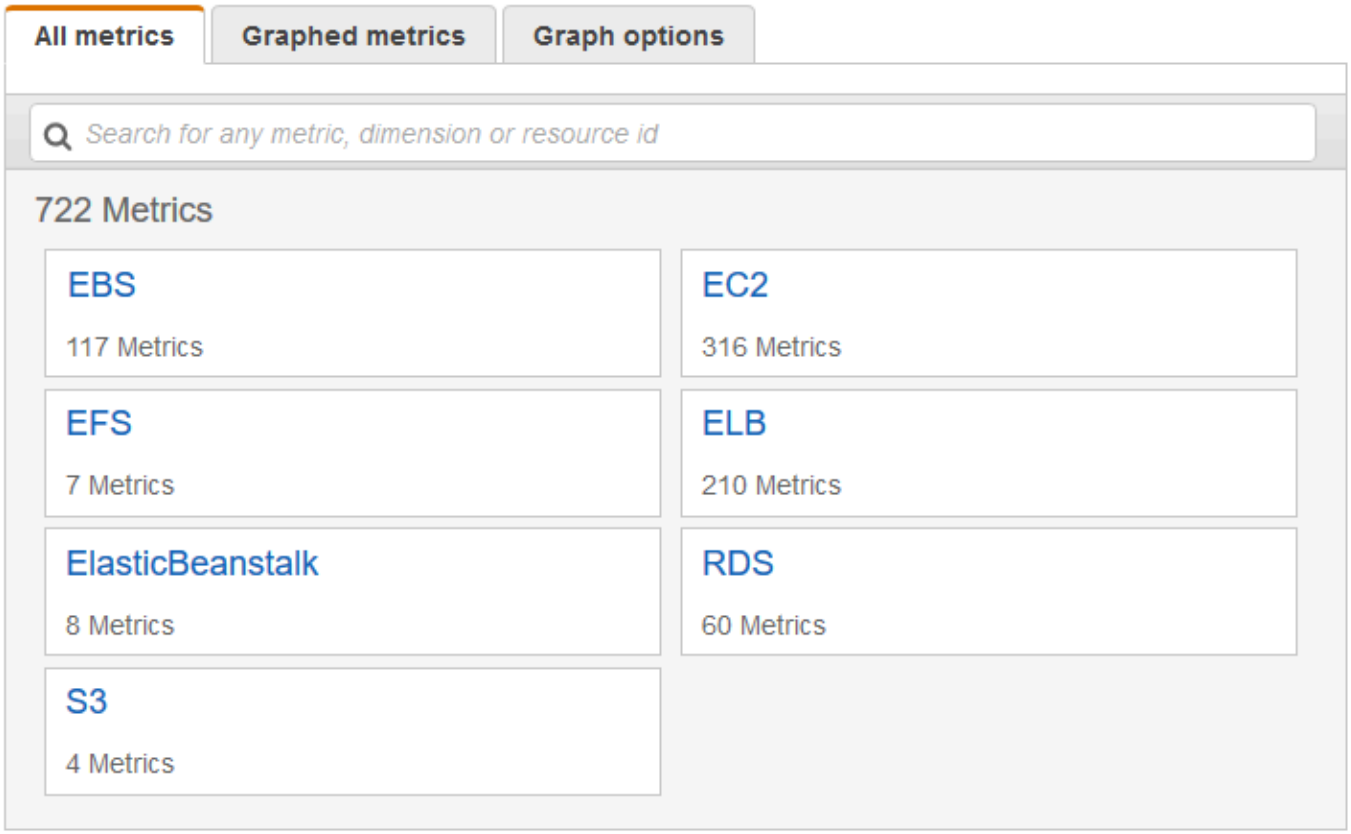
Das folgende Beispiel zeigt, wie Sie die maximale CPU-Auslastung bei einer bestimmten EC2 Instance bestimmen.

Voraussetzungen

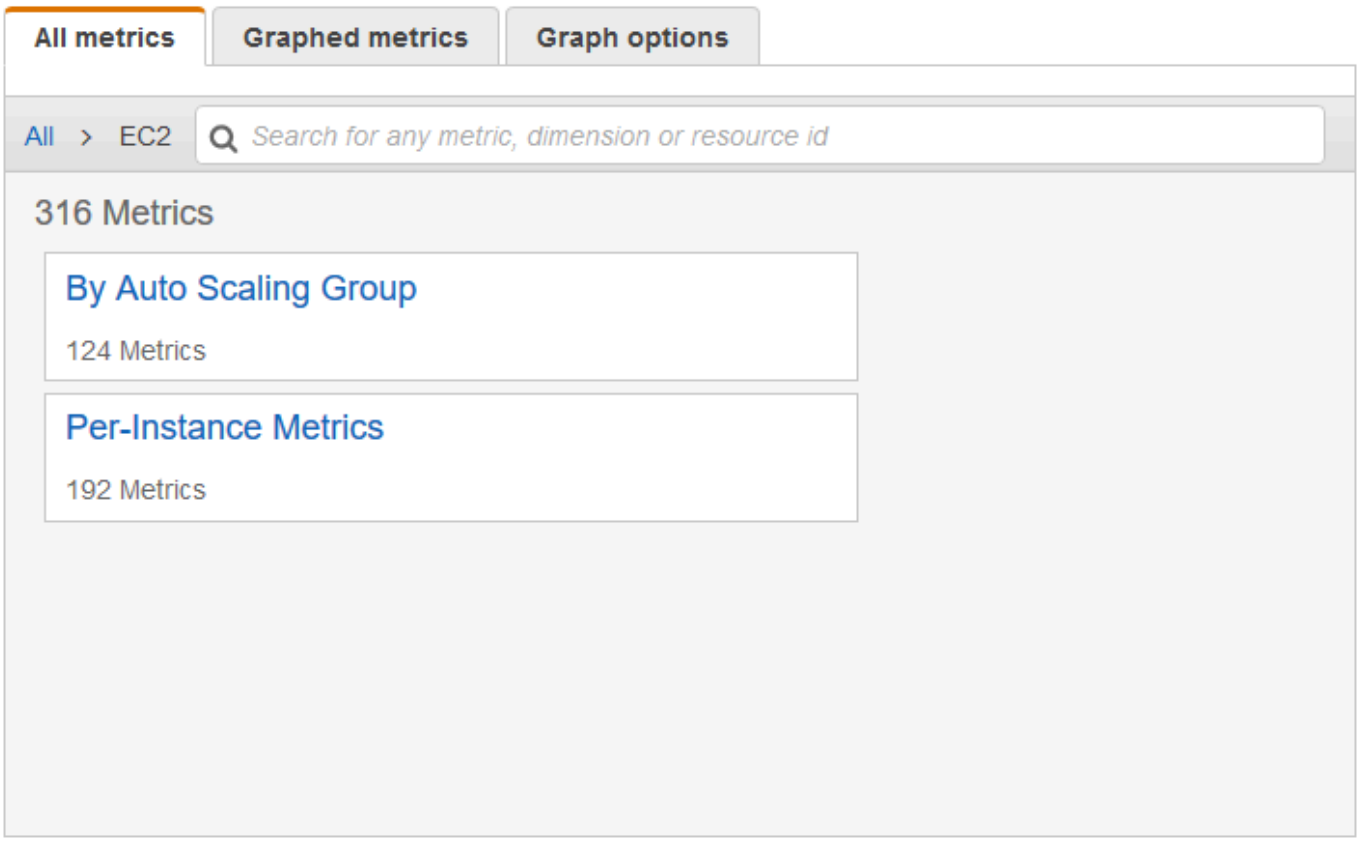
- Sie müssen die ID der Instance kennen. Sie können die Instance-ID mit der Amazon-EC2-Konsole oder mit dem Befehl describe-instances abrufen.
- Standardmäßig ist die grundlegende Überwachung aktiviert. Sie können aber auch eine detaillierte Überwachung aktivieren. Weitere Informationen finden Sie unter Aktivieren oder deaktivieren Sie die detaillierte Überwachung für Ihre Instances im Amazon-EC2-Benutzerhandbuch für Linux-Instances.

So zeigen Sie die durchschnittliche CPU-Auslastung für eine bestimmte Instance mithilfe der Konsole an:

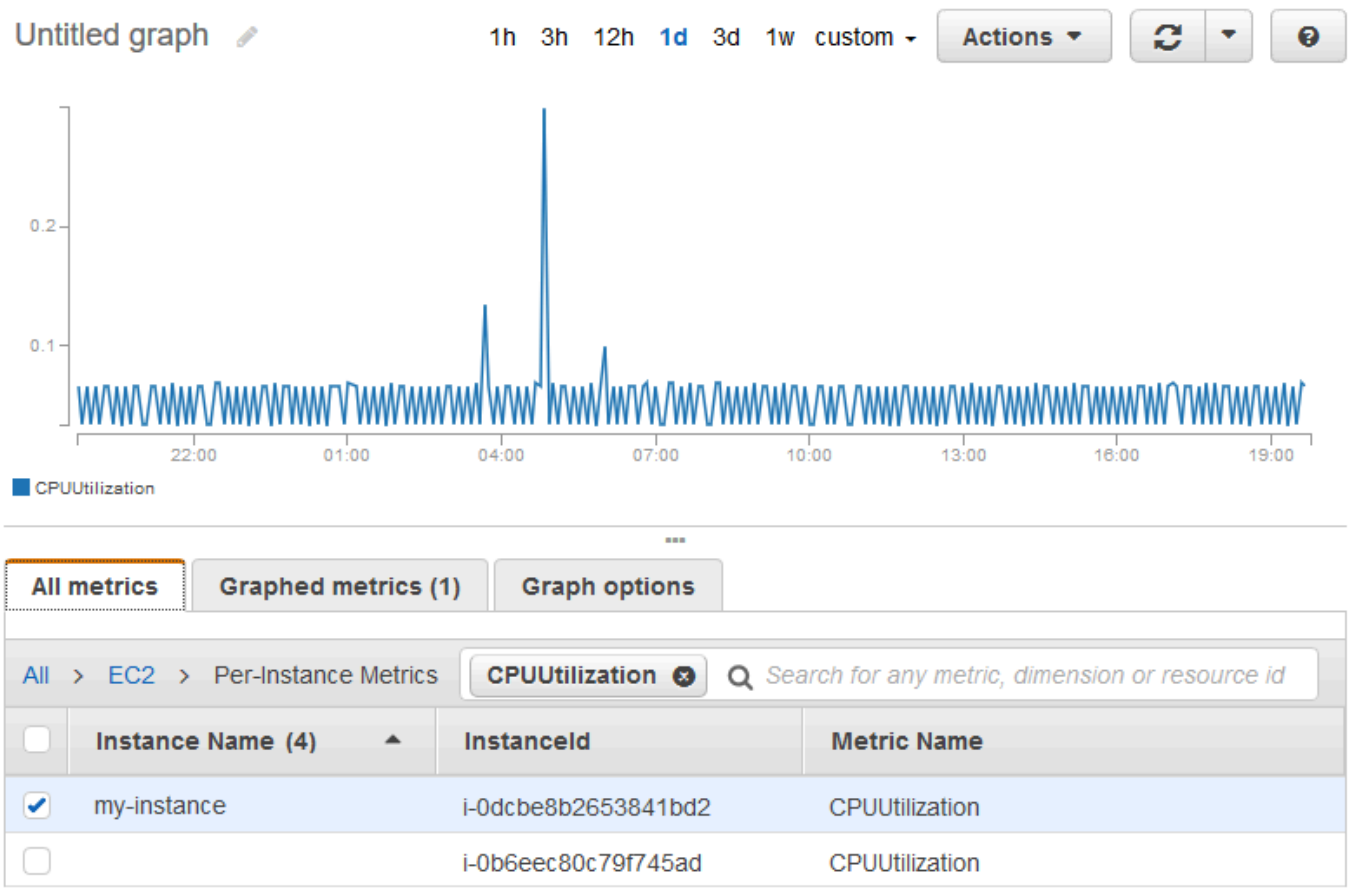
1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie den Namespace der EC2-Metrik aus.



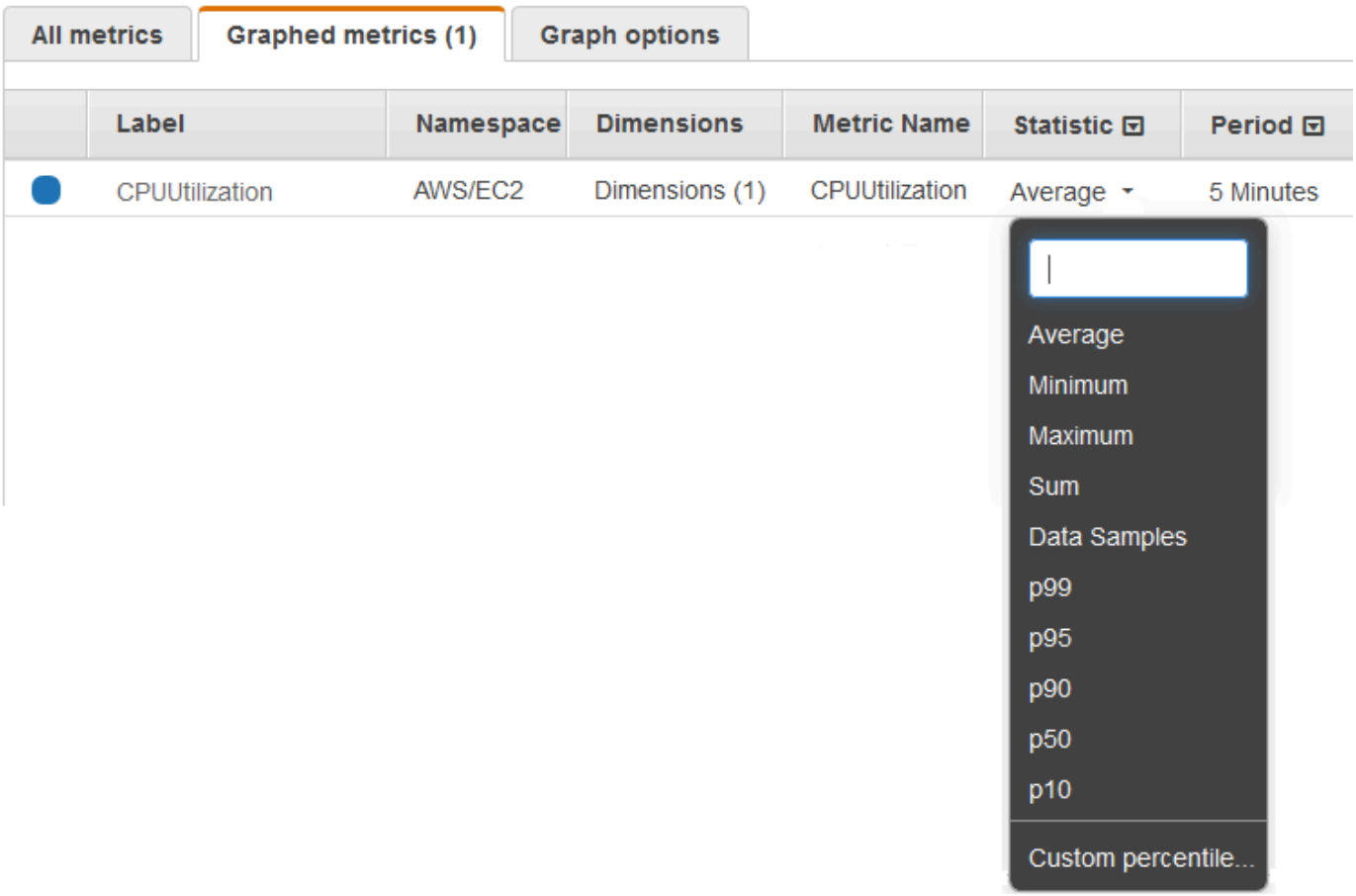
4. Wählen Sie die Dimension Per-Instance Metrics (Metriken pro Instance) aus.



5. Geben Sie **`CPUUtilization`** in das Suchfeld ein und drücken Sie die Eingabetaste. Wählen Sie die Zeile für eine Instance aus, in der ein Diagramm mit der `CPUUtilization`-Metrik für die Instance angezeigt wird. Wenn Sie den Namen des Diagramms ändern möchten, wählen Sie das Bleistiftsymbol. Wenn Sie den Zeitraum ändern möchten, müssen Sie einen der vordefinierten Werte oder custom (benutzerdefiniert) auswählen.



6. Wenn Sie die Statistik ändern möchten, wählen Sie die Registerkarte Graphed metrics aus. Wählen Sie die Spaltenüberschrift oder einen einzelnen Wert aus, und wählen Sie dann eine der Statistiken oder vordefinierten Perzentile aus, oder geben Sie eine benutzerdefinierte Perzentile ein (z. B. **p99.999**).



7. Wenn Sie den Zeitraum ändern möchten, wählen Sie die Registerkarte Graphed metrics aus. Wählen Sie die Spaltenüberschrift oder einen einzelnen Wert und anschließend einen anderen Wert aus.

Um die CPU-Auslastung pro EC2-Instance zu ermitteln, verwenden Sie AWS CLI

Verwenden Sie den get-metric-statisticsBefehl wie folgt, um die `CPUUtilization` Metrik für die angegebene Instance abzurufen.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization \
--dimensions Name=InstanceId,Value=i-1234567890abcdef0 --statistics Maximum \
--start-time 2016-10-18T23:18:00 --end-time 2016-10-19T23:18:00 --period 360
```

Die zurückgegebenen Statistiken sind Sechs-Minuten-Werte für das angeforderte 24-Stunden-Intervall. Jeder Wert stellt die maximale CPU-Auslastung in Prozent für die angegebene Instance für einen bestimmten Zeitraum von 6 Minuten dar. Beachten Sie, dass die Datenpunkte nicht in chronologischer Reihenfolge zurückgegeben werden. Das folgende Beispiel zeigt den Anfang

der Ausgabe (die vollständige Ausgabe umfasst Datenpunkte für alle 6 Minuten des 24-Stunden-Zeitraums).

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-10-19T00:18:00Z",
            "Maximum": 0.33000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-19T03:18:00Z",
            "Maximum": 99.670000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-19T07:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```

Statistiken zwischen Ressourcen aggregieren

Sie können die Metriken für AWS Ressourcen über mehrere Ressourcen hinweg aggregieren. Metriken sind zwischen Regionen vollständig getrennt, aber Sie können Metrikberechnungen verwenden, um ähnliche Metriken über Regionen hinweg zu aggregieren. Weitere Informationen finden Sie unter Verwenden von Metrikberechnungen.

Sie können beispielsweise Statistiken für Ihre EC2 Instances aggregieren, für die die detaillierte Überwachung aktiviert ist. Instances, die die grundlegende Überwachung verwenden, sind nicht enthalten. Daher müssen Sie detaillierte Überwachung (gegen eine zusätzliche Gebühr) aktivieren. Damit werden Daten in 1-Minuten-Intervallen bereitgestellt. Weitere Informationen finden Sie unter Aktivieren oder deaktivieren Sie die detaillierte Überwachung für Ihre Instances im Amazon-EC2-Benutzerhandbuch für Linux-Instances.

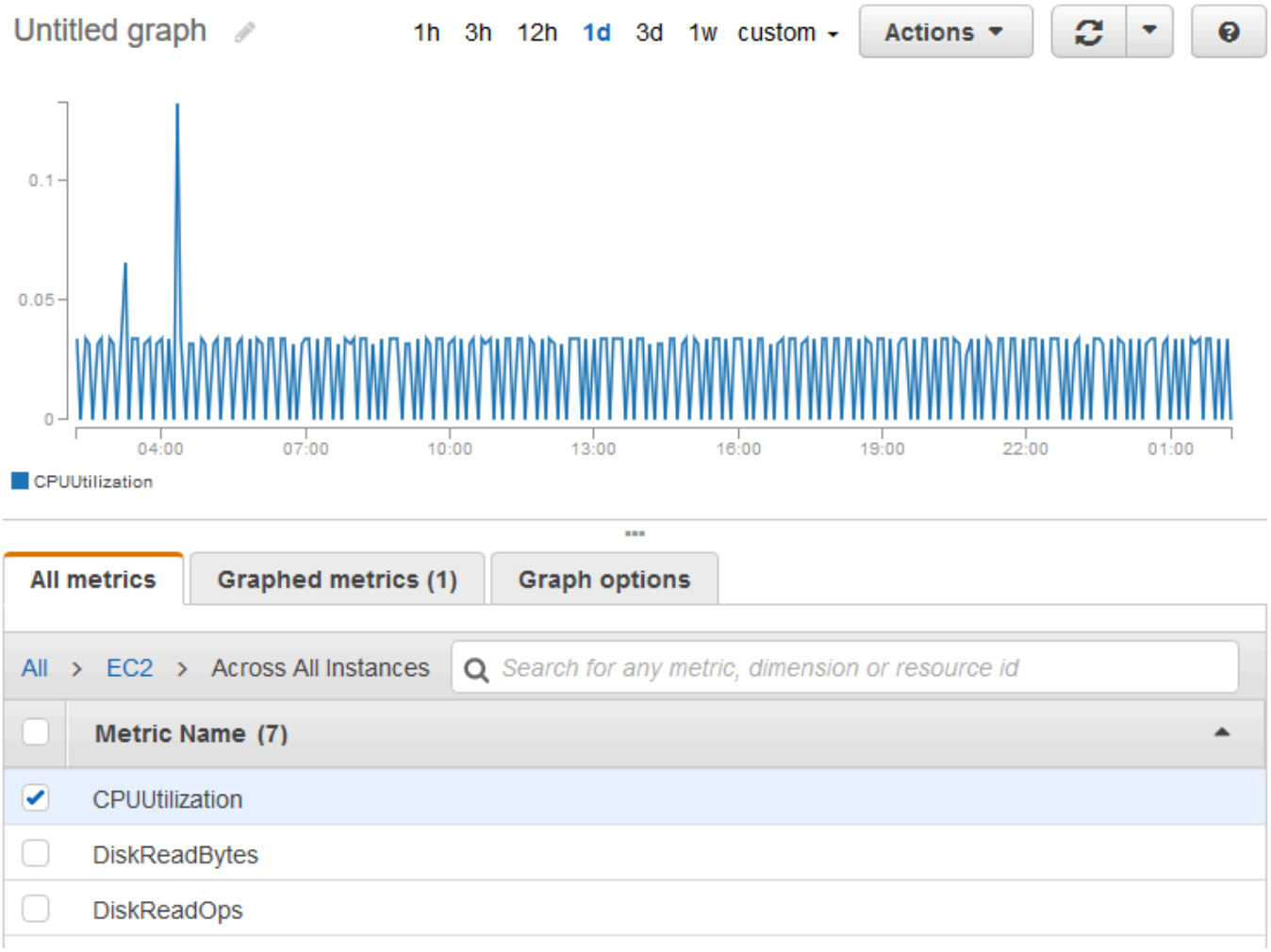
Dieses Beispiel zeigt, wie Sie die durchschnittliche CPU-Auslastung für Ihre EC2 Instances abrufen. Da keine Dimension angegeben ist, werden Statistiken für alle Dimensionen im `AWS/EC2`

Namespace CloudWatch zurückgegeben. Informationen zum Abrufen von Statistiken für andere Metriken finden Sie unter AWS Dienste, die CloudWatch Metriken veröffentlichen.

> **Important**
>
> Diese Technik zum Abrufen aller Dimensionen in einem AWS Namespace funktioniert nicht für benutzerdefinierte Namespaces, in denen Sie veröffentlichen. CloudWatch Bei benutzerdefinierten Namespaces müssen Sie die vollständige Palette von Dimensionen angeben, die im Zusammenhang mit einem bestimmten Datenpunkt stehen, um Statistiken zu diesem Datenpunkt abzurufen.

So zeigen Sie die durchschnittliche CPU-Auslastung für Ihre EC2 Instances an

1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/cloudwatch/. CloudWatch
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie den Namespace EC2 und Across All Instances aus.
4. Wählen Sie die Zeile mit `CPUUtilization` aus, in der ein Diagramm mit der Metrik für alle EC2 Instances angezeigt wird. Wenn Sie den Namen des Diagramms ändern möchten, wählen Sie das Bleistiftsymbol. Wenn Sie den Zeitraum ändern möchten, müssen Sie einen der vordefinierten Werte oder custom (benutzerdefiniert) auswählen.



5. Wenn Sie die Statistik ändern möchten, wählen Sie die Registerkarte Graphed metrics aus. Wählen Sie die Spaltenüberschrift oder einen einzelnen Wert aus, und wählen Sie dann eine der Statistiken oder vordefinierten Perzentile aus, oder geben Sie eine benutzerdefinierte Perzentile ein (z. B. **`p95.45`**).
6. Wenn Sie den Zeitraum ändern möchten, wählen Sie die Registerkarte Graphed metrics aus. Wählen Sie die Spaltenüberschrift oder einen einzelnen Wert aus, und wählen Sie dann einen anderen Wert aus.

Um die durchschnittliche CPU-Auslastung Ihrer EC2-Instances zu ermitteln, verwenden Sie AWS CLI

Verwenden Sie den get-metric-statistics-Befehl wie folgt:

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization
 --statistics "Average" "SampleCount" \
--start-time 2016-10-11T23:18:00 --end-time 2016-10-12T23:18:00 --period 3600
```

Das Folgende ist Ausgabebeispiel:

```
{
    "Datapoints": [
        {
            "SampleCount": 238.0,
            "Timestamp": "2016-10-12T07:18:00Z",
            "Average": 0.038235294117647062,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2016-10-12T09:18:00Z",
            "Average": 0.16670833333333332,
            "Unit": "Percent"
        },
        {
            "SampleCount": 238.0,
            "Timestamp": "2016-10-11T23:18:00Z",
            "Average": 0.041596638655462197,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```

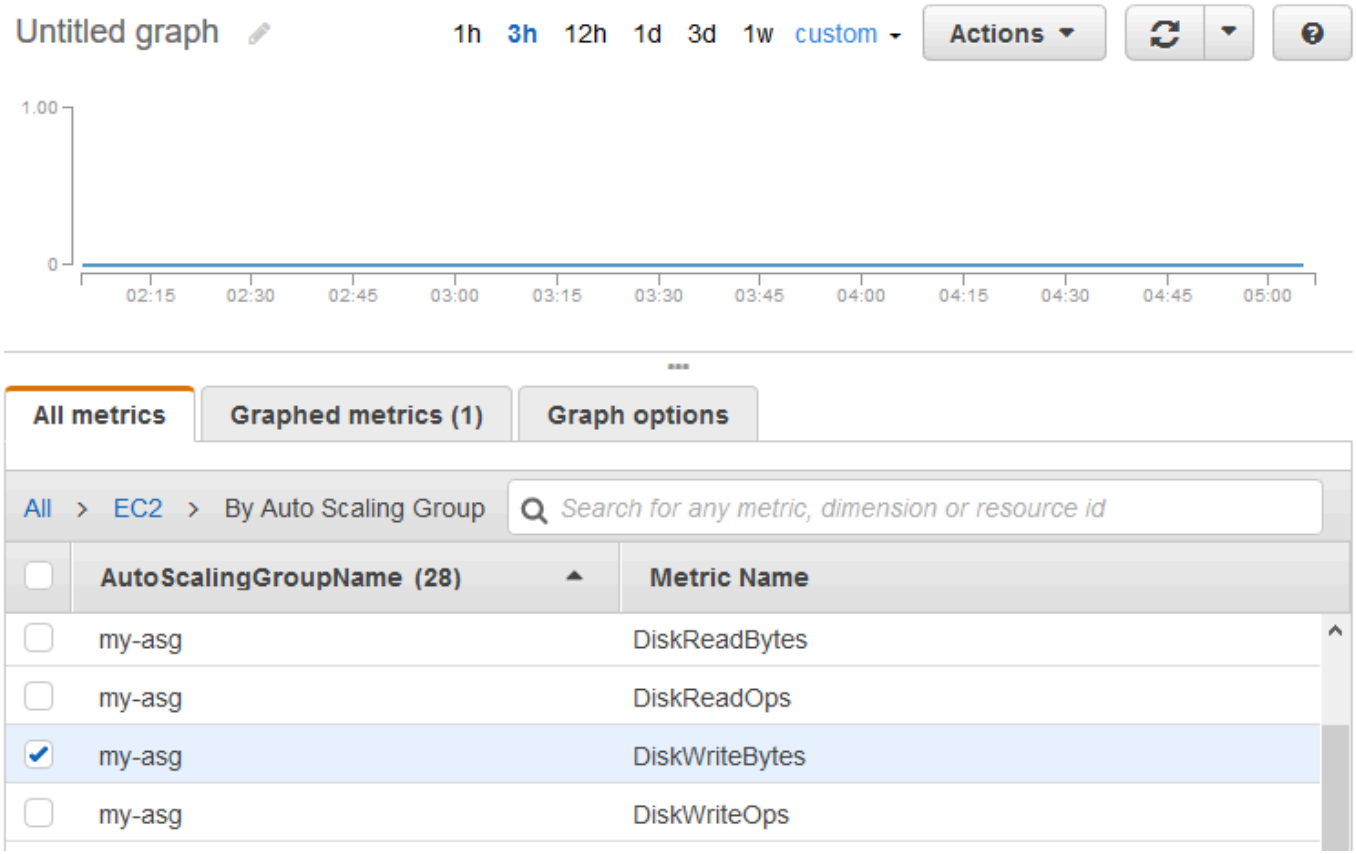
Aggregieren von Statistiken nach Auto Scaling-Gruppe

Sie können Statistiken für die EC2 Instances in einer Auto Scaling-Gruppe aggregieren. Die Metriken sind zwischen den Regionen völlig getrennt, aber Sie können CloudWatch Metrikmathematik verwenden, um Metriken aus mehreren Regionen zu aggregieren und zu transformieren. Sie können auch das kontoübergreifende Dashboard verwenden, um Metrikberechnungen für Metriken verschiedener Konten durchzuführen.

Dieses Beispiel zeigt, wie Sie die Gesamtzahl der Bytes anzeigen, die für eine einzelne Auto-Scaling-Gruppe auf die Festplatte geschrieben werden. Der Gesamtzahl wird für einminütige Zeiträume eines 24-Stunden-Intervalls für alle EC2 Instances in der angegebenen Auto Scaling-Gruppe berechnet.

So zeigen Sie DiskWriteBytes die Instances in einer Auto Scaling Scaling-Gruppe mithilfe der Konsole an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie den Namespace EC2 und anschließend By Auto Scaling Group (Nach Auto Scaling-Gruppe) aus.
4. Wählen Sie die Zeile für die DiskWriteBytesMetrik und die spezifische Auto Scaling Scaling-Gruppe aus, in der ein Diagramm für die Metrik für die Instances in der Auto Scaling Scaling-Gruppe angezeigt wird. Wenn Sie den Namen des Diagramms ändern möchten, wählen Sie das Bleistiftsymbol. Wenn Sie den Zeitraum ändern möchten, müssen Sie einen der vordefinierten Werte oder custom (benutzerdefiniert) auswählen.



5. Wenn Sie die Statistik ändern möchten, wählen Sie die Registerkarte Graphed metrics aus. Wählen Sie die Spaltenüberschrift oder einen einzelnen Wert aus, und wählen Sie dann eine der Statistiken oder vordefinierten Perzentile aus, oder geben Sie eine benutzerdefinierte Perzentile ein (z. B. **`p95.45`**).

6. Wenn Sie den Zeitraum ändern möchten, wählen Sie die Registerkarte Graphed metrics aus. Wählen Sie die Spaltenüberschrift oder einen einzelnen Wert aus, und wählen Sie dann einen anderen Wert aus.

Um DiskWriteBytes nach den Instances in einer Auto Scaling Scaling-Gruppe zu suchen, verwenden Sie den AWS CLI

Verwenden Sie den get-metric-statistics-Befehl wie folgt:

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name DiskWriteBytes
--dimensions Name=AutoScalingGroupName,Value=my-asg --statistics "Sum" "SampleCount" \
--start-time 2016-10-16T23:18:00 --end-time 2016-10-18T23:18:00 --period 360
```

Es folgt eine Beispielausgabe.

```
{
    "Datapoints": [
        {
            "SampleCount": 18.0,
            "Timestamp": "2016-10-19T21:36:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "SampleCount": 5.0,
            "Timestamp": "2016-10-19T21:42:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        }
    ],
    "Label": "DiskWriteBytes"
}
```

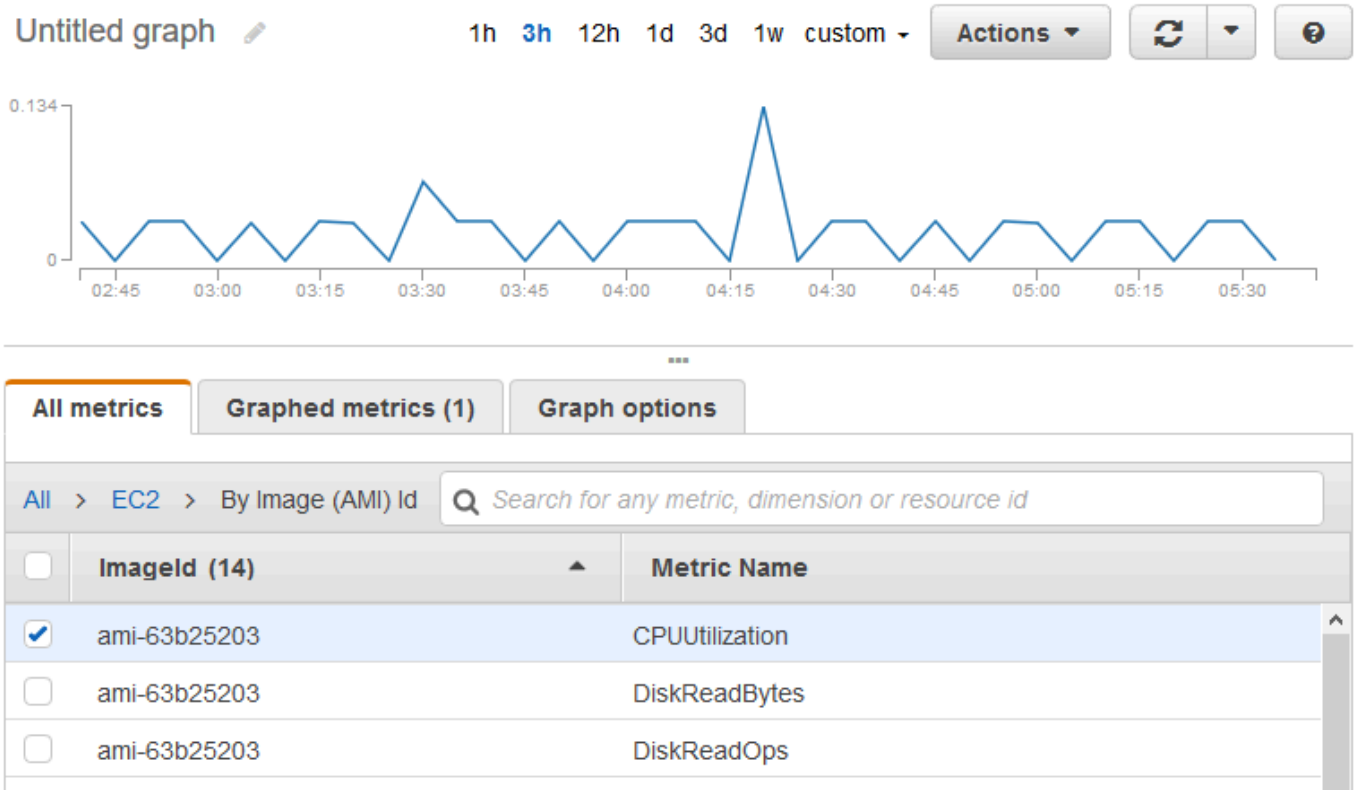
Statistiken nach Amazon Machine Image (AMI) aggregieren

Sie können Statistiken für die EC2 Instances aggregieren, für die die detaillierte Überwachung aktiviert ist. Instances, die die grundlegende Überwachung verwenden, sind nicht enthalten. Weitere Informationen finden Sie unter Aktivieren oder deaktivieren Sie die detaillierte Überwachung für Ihre Instances im Amazon-EC2-Benutzerhandbuch für Linux-Instances.

Dieses Beispiel zeigt, wie Sie die durchschnittliche CPU-Auslastung für alle Instances, die das angegebene AMI verwenden, abrufen. Der Durchschnitt wird über 60-Sekunden-Intervalle für einen Zeitraum von einem Tag berechnet.

So zeigen Sie die durchschnittliche CPU-Auslastung nach AMI mithilfe der Konsole an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie den Namespace EC2 aus und dann By Image (AMI) Id (Nach Image-ID (AMI)).
4. Wählen Sie die Zeile für die `CPUUtilization`-Metrik und das spezifische AMI aus, das ein Diagramm für die Metrik für ein bestimmtes AMI anzeigt. Wenn Sie den Namen des Diagramms ändern möchten, wählen Sie das Bleistiftsymbol. Wenn Sie den Zeitraum ändern möchten, müssen Sie einen der vordefinierten Werte oder custom (benutzerdefiniert) auswählen.



5. Wenn Sie die Statistik ändern möchten, wählen Sie die Registerkarte Graphed metrics aus. Wählen Sie die Spaltenüberschrift oder einen einzelnen Wert aus, und wählen Sie dann eine der Statistiken oder vordefinierten Perzentile aus, oder geben Sie eine benutzerdefinierte Perzentile ein (z. B. **`p95.45`**).

6. Wenn Sie den Zeitraum ändern möchten, wählen Sie die Registerkarte Graphed metrics aus. Wählen Sie die Spaltenüberschrift oder einen einzelnen Wert aus, und wählen Sie dann einen anderen Wert aus.

Um die durchschnittliche CPU-Auslastung durch AMI zu ermitteln, verwenden Sie AWS CLI

Verwenden Sie den get-metric-statistics-Befehl wie folgt:

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization \
--dimensions Name=ImageId,Value=ami-3c47a355 --statistics Average \
--start-time 2016-10-10T00:00:00 --end-time 2016-10-11T00:00:00 --period 3600
```

Die Operation gibt die Statistiken in Ein-Stunden-Werten für ein Intervall von einem Tag zurück. Jeder Wert stellt die durchschnittliche CPU-Auslastung in Prozent für EC2 Instances mit dem angegebenen AMI dar. Es folgt eine Beispielausgabe.

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-10-10T07:00:00Z",
            "Average": 0.041000000000000009,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-10T14:00:00Z",
            "Average": 0.079579831932773085,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-10T06:00:00Z",
            "Average": 0.036000000000000011,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```

Veröffentlichen von benutzerdefinierten -Metriken

Sie können Ihre eigenen Metriken CloudWatch mithilfe der AWS CLI oder einer API veröffentlichen. Mit dem können Sie statistische Grafiken Ihrer veröffentlichten Metriken anzeigen AWS Management Console.

CloudWatch speichert Daten zu einer Metrik als eine Reihe von Datenpunkten. Jeder Datenpunkt verfügt über einen zugewiesenen Zeitstempel. Sie können auch eine aggregierte Gruppe von Datenpunkten, die sogenannte Statistikgruppe veröffentlichen.

Themen

Hochauflösende Metriken

Jede Metrik entspricht einer der folgenden:

- Standardauflösung; hierbei haben die Daten eine Granularität von einer Minute
- Hohe Auflösung; hierbei haben die Daten eine Granularität von einer Sekunde

Von AWS Diensten erzeugte Metriken haben standardmäßig eine Standardauflösung. Wenn Sie eine benutzerdefinierte Metrik veröffentlichen, hat diese entweder die Standardauflösung oder eine hohe Auflösung. Wenn Sie eine Metrik mit hoher Auflösung veröffentlichen, wird sie mit einer Auflösung von 1 Sekunde CloudWatch gespeichert, sodass Sie sie mit einem Zeitraum von 1 Sekunde, 5 Sekunden, 10 Sekunden, 30 Sekunden oder einem beliebigen Vielfachen von 60 Sekunden lesen und abrufen können.

Mit hochauflösenden Metriken erhalten Sie genauere Einblicke in die Aktivitäten Ihrer Anwendung, die unter einer Minute liegen. Denken Sie daran, dass jeder `PutMetricData`-Aufruf einer benutzerdefinierten Metrik in Rechnung gestellt wird, sodass höhere Gebühren entstehen

können, wenn Sie häufiger `PutMetricData`-Aufrufe hochauflösender Metrik ausführen. Weitere Informationen zur CloudWatch Preisgestaltung finden Sie unter CloudWatch Amazon-Preise.

Wenn Sie einen Alarm für eine hochauflösende Metrik festlegen, können Sie einen hochauflösenden Alarm für einen Zeitraum von 10 Sekunden oder 30 Sekunden oder einen regelmäßigen Alarm für einen Zeitraum festlegen, der ein Mehrfaches von 60 Sekunden beträgt. Die Gebühr für hochauflösende Alarme mit einem Zeitraum von 10 oder 30 Sekunden ist höher.

Dimensionen verwenden

In benutzerdefinierten Metriken, wird der Parameter `--dimensions` häufig verwendet. Eine Dimension macht außerdem deutlich, um welche Metrik es sich handelt und welche Daten darin gespeichert werden. Sie können einer Metrik bis zu 30 Dimensionen zuweisen, und jede Dimension wird durch ein Name-Wert-Paar definiert.

Die Art und Weise, wie Sie eine Dimension angeben, ist bei der Verwendung verschiedener Befehle unterschiedlich. Mit put-metric-datageben Sie jede Dimension als *`MyName`*= an *`MyValue`*, und mit get-metric-statisticsoder verwenden put-metric-alarmSie das Format `Name=` *`MyName`*, `Value=` *`MyValue`*. Mit dem folgenden Befehl wird eine `Buffers`-Metrik mit zwei Dimensionen namens `InstanceId` und `InstanceType` veröffentlicht.

```
aws cloudwatch put-metric-data --metric-name Buffers --namespace MyNameSpace --unit
 Bytes --value 231434333 --dimensions InstanceId=1-23456789,InstanceType=m1.small
```

Dieser Befehl ruft die Statistiken für dieselbe Metrik ab. Trennen Sie Namen und Wert einer einzelnen Dimension durch Kommata voneinander ab. Wenn Sie mit mehreren Dimensionen arbeiten, verwenden Sie zwischen einer Dimension und der nächsten ein Leerzeichen.

```
aws cloudwatch get-metric-statistics --metric-name Buffers --namespace MyNameSpace --
dimensions Name=InstanceId,Value=1-23456789 Name=InstanceType,Value=m1.small --start-
time 2016-10-15T04:00:00Z --end-time 2016-10-19T07:00:00Z --statistics Average --period
 60
```

Wenn eine einzelne Metrik mehrere Dimensionen umfasst, müssen Sie bei der Verwendung einen Wert für jede definierte Dimension angeben get-metric-statistics. Die Amazon S3 S3-Metrik `BucketSizeBytes` umfasst beispielsweise die Dimensionen `BucketName` und`StorageType`, sodass Sie beide Dimensionen mit angeben müssen get-metric-statistics.

```
aws cloudwatch get-metric-statistics --metric-name BucketSizeBytes --start-time
 2017-01-23T14:23:00Z --end-time 2017-01-26T19:30:00Z --period 3600 --namespace
```

```
AWS/S3 --statistics Maximum --dimensions Name=BucketName,Value=MyBucketName
Name=StorageType,Value=StandardStorage --output table
```

Um die für eine Metrik definierten Dimensionen anzuzeigen, verwenden Sie den Befehl list-metrics.

Einzelne Datenpunkte veröffentlichen

Um einen einzelnen Datenpunkt für eine neue oder bestehende Metrik zu veröffentlichen, verwenden Sie den put-metric-dataBefehl mit einem Wert und einem Zeitstempel. Die folgenden Aktionen veröffentlichen z. B. jeweils einen Datenpunkt.

```
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService --
value 2 --timestamp 2016-10-20T12:00:00.000Z
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService --
value 4 --timestamp 2016-10-20T12:00:01.000Z
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService --
value 5 --timestamp 2016-10-20T12:00:02.000Z
```

Wenn Sie diesen Befehl mit einem neuen Metriknamen aufrufen, CloudWatch wird eine Metrik für Sie erstellt. Ordnet CloudWatch andernfalls Ihre Daten der vorhandenen Metrik zu, die Sie angegeben haben.

> **Note**
>
> Wenn Sie eine Metrik erstellen, kann es bis zu 2 Minuten dauern, bis Sie mit dem get-metric-statisticsBefehl Statistiken für die neue Metrik abrufen können. Es kann jedoch bis zu 15 Minuten dauern, bevor die neue Metrik in der Liste der Metriken angezeigt wird, die mit dem Befehl list-metrics abgerufen wird.

Sie können zwar Datenpunkte mit Zeitstempeln veröffentlichen, die bis zu einer Tausendstelsekunde genau sind, CloudWatch aggregiert die Daten jedoch auf eine Mindestgranularität von 1 Sekunde. CloudWatch zeichnet den Durchschnitt (Summe aller Elemente geteilt durch die Anzahl der Elemente) der für jeden Zeitraum erhaltenen Werte sowie die Anzahl der Stichproben, den Höchstwert und den Minimalwert für denselben Zeitraum auf. Die Metrik `PageViewCount` aus den vorherigen Beispielen enthält z. B. drei Datenpunkte mit Zeitstempeln im Abstand von wenigen Sekunden. Wenn Sie Ihren Zeitraum auf 1 Minute festgelegt haben, werden die drei Datenpunkte CloudWatch aggregiert, da sie alle Zeitstempel innerhalb eines Zeitraums von 1 Minute haben.

Mit dem Befehl get-metric-statistics können Sie Statistiken basierend auf den veröffentlichten Datenpunkten abrufen.

```
aws cloudwatch get-metric-statistics --namespace MyService --metric-name PageViewCount
 \
--statistics "Sum" "Maximum" "Minimum" "Average" "SampleCount" \
--start-time 2016-10-20T12:00:00.000Z --end-time 2016-10-20T12:05:00.000Z --period 60
```

Es folgt eine Beispielausgabe.

```
{
    "Datapoints": [
        {
            "SampleCount": 3.0,
            "Timestamp": "2016-10-20T12:00:00Z",
            "Average": 3.6666666666666665,
            "Maximum": 5.0,
            "Minimum": 2.0,
            "Sum": 11.0,
            "Unit": "None"
        }
    ],
    "Label": "PageViewCount"
}
```

Statistikgruppen veröffentlichen

Sie können Ihre Daten vor dem Veröffentlichen in aggregieren. CloudWatch Bei mehreren Datenpunkten pro Minute wird die Anzahl der Aufrufe für put-metric-data durch die Aggregierung der Daten minimiert. Anstatt zum Beispiel den Befehl put-metric-data mehrfach für drei Datenpunkte aufzurufen, die in einem Abstand von drei Sekunden zueinander liegen, können Sie die Daten in einer Statistikgruppe zusammenfassen (= aggregieren), die Sie mit einem Aufruf mit dem Parameter `--statistic-values` veröffentlichen.

```
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService
 --statistic-values Sum=11,Minimum=2,Maximum=5,SampleCount=3 --
timestamp 2016-10-14T12:00:00.000Z
```

CloudWatch benötigt Rohdatenpunkte, um Perzentile zu berechnen. Wenn Sie Daten stattdessen mit einer Statistikgruppe veröffentlichen, können Sie nur dann eine Perzentil-Statistik für diese Daten abrufen, wenn eine der folgenden Bedingungen erfüllt ist:

- Der `SampleCount` der Statistikgruppe ist 1.
- `Minimum` und `Maximum` der Statistikgruppe sind gleich.

Den Nullwert veröffentlichen

Wenn Ihre Daten seltener erfasst werden und es Zeiträume ohne verknüpfte Daten gibt, können Sie den Wert Null (`0`) für diesen Zeitraum oder gar keinen Wert veröffentlichen. Wenn Sie durch regelmäßige Aufrufe an `PutMetricData` den Zustand Ihrer Anwendung überwachen, können Sie Null anstelle von gar keinem Wert veröffentlichen. Sie können beispielsweise einen CloudWatch Alarm einrichten, der Sie benachrichtigt, wenn Ihre Anwendung nicht alle fünf Minuten Messwerte veröffentlicht. Eine solche Anwendung soll für die Zeiträume ohne verknüpfte Daten Nullen veröffentlichen.

Sie können auch Nullen veröffentlichen, wenn Sie die Gesamtzahl der Datenpunkte nachverfolgen möchten, oder um Statistiken, wie z. B. Mindest- und durchschnittliche Datenpunkte, den Wert "0" enthalten sollen.

Veröffentlichen von Metriken beenden

Um die Veröffentlichung benutzerdefinierter Metriken zu beenden CloudWatch, ändern Sie den Code Ihrer Anwendung oder Ihres Dienstes so, dass er nicht mehr verwendet wird PutMetricData. CloudWatch ruft keine Metriken aus Anwendungen ab, sondern empfängt nur das, was an ihn weitergeleitet wird. Um also die Veröffentlichung Ihrer Metriken zu beenden, müssen Sie sie an der Quelle stoppen.

CloudWatch Amazon-Alarme verwenden

In Amazon können Sie metrische und zusammengesetzte Alarme erstellen CloudWatch.

- Ein metrischer Alarm überwacht eine einzelne CloudWatch Metrik oder das Ergebnis eines mathematischen Ausdrucks, der auf CloudWatch Metriken basiert. Der Alarm führt eine oder mehrere Aktionen durch, die vom Wert der Metrik oder des Ausdrucks im Vergleich zu einem Schwellenwert in einer Reihe von Zeiträumen abhängt. Die Aktion kann das Senden einer Benachrichtigung an ein Amazon SNS SNS-Thema, das Ausführen einer Amazon EC2-Aktion oder einer Amazon EC2 Auto Scaling Scaling-Aktion oder das Erstellen eines OR-Vorfalls in OpsItem Systems Manager sein.
- Ein zusammengesetzter Alarm enthält einen Regelausdruck, der die Alarmstatus anderer Alarme, die Sie erstellt haben, berücksichtigt. Der zusammengesetzte Alarm geht nur dann in den ALARM-Status über, wenn alle Bedingungen der Regel erfüllt sind. Die im Regelausdruck eines zusammengesetzten Alarms angegebenen Alarme können Metrikalarme und andere zusammengesetzte Alarme umfassen.

  Die Verwendung zusammengesetzter Alarme kann das Alarmrauschen reduzieren. Sie können mehrere metrische Alarme erstellen und auch einen zusammengesetzten Alarm erstellen und Alarme nur für den zusammengesetzten Alarm einrichten. Beispielsweise kann ein zusammengesetzter Alarm nur dann in den ALARM-Status übergehen, wenn sich alle zugrunde liegenden Metrikalarme im ALARM-Status befinden.

  Zusammengesetzte Alarme können Amazon SNS SNS-Benachrichtigungen senden, wenn sie ihren Status ändern, und können Systems Manager OpsItems oder Incidents auslösen, wenn sie in den ALARM-Status wechseln, können aber keine EC2-Aktionen oder Auto Scaling Scaling-Aktionen ausführen.

> **Note**
>
> Sie können in Ihrem AWS Konto so viele Alarme erstellen, wie Sie möchten.

Sie können Alarme zu Dashboards hinzufügen, sodass Sie Ihre AWS Ressourcen und Anwendungen in mehreren Regionen überwachen und Benachrichtigungen darüber erhalten können. Nachdem Sie einen Alarm zu einem Dashboard hinzugefügt haben, wird der Alarm grau, wenn er sich im

INSUFFICIENT_DATA-Zustand befindet, und rot, wenn er sich im ALARM-Zustand befindet. Der Alarm wird ohne Farbe angezeigt, wenn er sich im OK-Zustand befindet.

Sie können kürzlich besuchte Alarme auch über die Option Favoriten und zuletzt verwendete Alarme im Navigationsbereich der CloudWatch Konsole zu Favoriten hinzufügen. Die Option Favoriten und kürzlich aufgerufene enthält Spalten für Ihre bevorzugten Alarme und die zuletzt besuchten Alarme.

Ein Alarm ruft Aktionen nur auf, wenn sich der Status des Alarms ändert. Eine Ausnahme bilden Alarme mit Auto-Scaling-Aktionen. Bei Auto-Scaling-Aktionen ruft der Alarm die Aktion weiterhin einmal pro Minute auf, sodass der Alarm im neuen Zustand bleibt.

Ein Alarm kann eine Metrik im selben Konto beobachten. Wenn Sie die kontoübergreifende Funktion in Ihrer CloudWatch Konsole aktiviert haben, können Sie auch Alarme erstellen, die Messwerte anderer AWS Konten überwachen. Das Erstellen von kontoübergreifenden zusammengesetzten Alarmen wird nicht unterstützt. Das Erstellen von kontoübergreifenden Alarmen, die mathematische Ausdrücke verwenden, wird unterstützt, außer dass die ANOMALY_DETECTION_BAND-, INSIGHT_RULE- und SERVICE_QUOTA-Funktionen für kontoübergreifende Alarme nicht unterstützt werden.

> **Note**
>
> CloudWatch testet oder validiert die von Ihnen angegebenen Aktionen nicht und erkennt auch keine Amazon EC2 Auto Scaling- oder Amazon SNS-Fehler, die auf einen Versuch zurückzuführen sind, nicht existierende Aktionen aufzurufen. Stellen Sie sicher, dass die Alarmaktionen vorhanden sind.

Metrikalarm-Status

Ein Metrikalarm kann die folgenden Status aufweisen:

- OK – Die Metrik oder der Ausdruck liegt innerhalb des festgelegten Schwellenwerts.
- ALARM – Die Metrik oder der Ausdruck liegt außerhalb des festgelegten Schwellenwerts.
- INSUFFICIENT_DATA – Der Alarm wurde soeben gestartet; die Metrik ist nicht verfügbar oder es sind nicht genügend Daten verfügbar, damit die Metrik den Alarmstatus bestimmen kann.

Auswerten eines Alarms

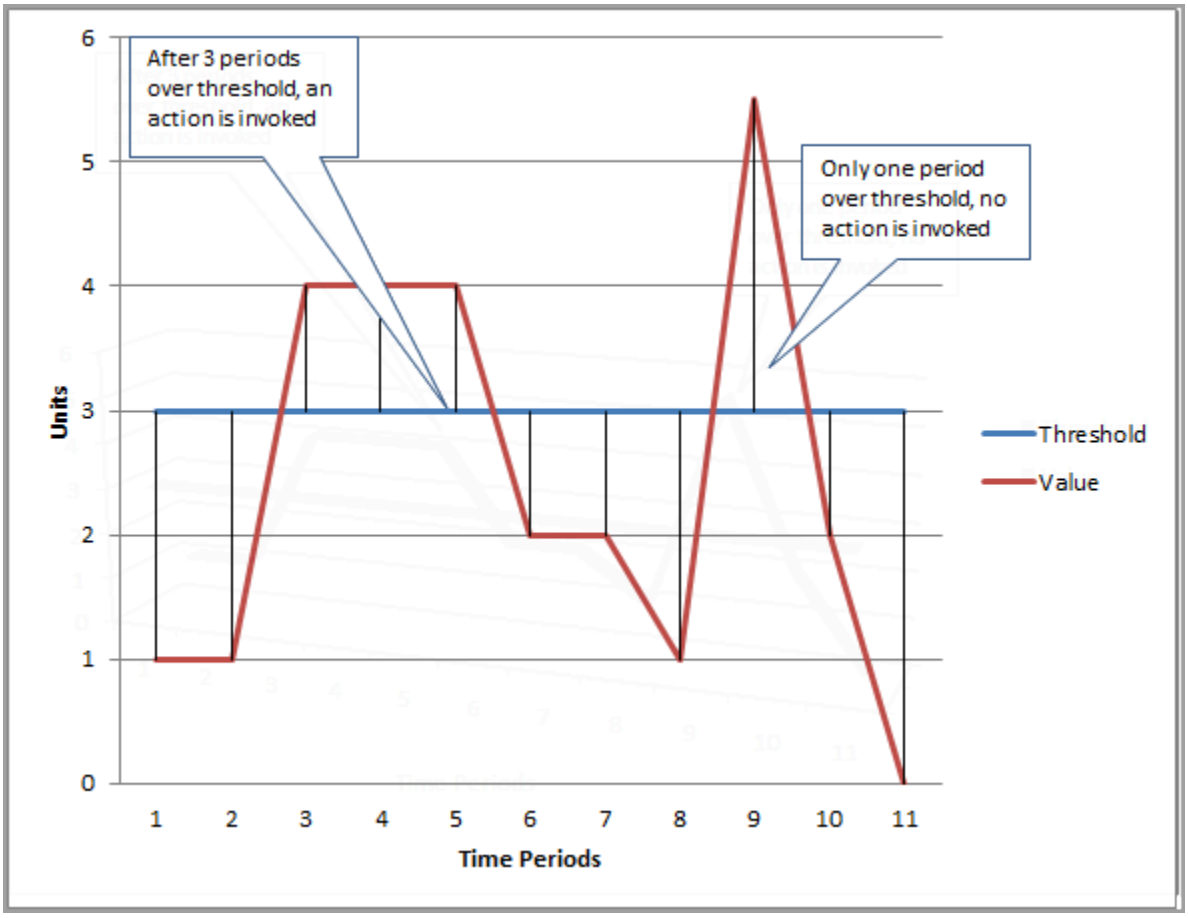
Wenn Sie einen Alarm erstellen, geben Sie drei Einstellungen an, anhand derer CloudWatch bewertet werden kann, wann der Alarmstatus geändert werden muss:

- Zeitraum ist die Zeitspanne, die für die Auswertung der Metrik oder des Ausdrucks verwendet wird, um jeden einzelnen Datenpunkt für einen Alarm zu erstellen. Sie wird in Sekunden angegeben.
- Evaluation Periods (Auswertungszeiträume) ist die Anzahl der neuesten Zeiträume, oder Datenpunkte, die beim Ermitteln des Alarmzustands zu bewerten sind.
- Datapoints to Alarm (Datenpunkte zum Alarm) ist die Anzahl der Datenpunkte innerhalb des Auswertungszeitraums, die überschritten werden müssen, um zu bewirken, dass der Alarm in den `ALARM`-Status versetzt wird. Die überschrittenen Datenpunkte müssen nicht aufeinanderfolgend sein, aber sie müssen alle innerhalb der letzten Datenpunkte liegen (dem Auswertungszeitraum entsprechend).

Für einen Zeitraum von einer Minute oder länger wird jede Minute ein Alarm ausgewertet, und die Auswertung basiert auf dem durch den Zeitraum und die Bewertungszeiträume definierten Zeitfenster. Wenn der Zeitraum beispielsweise 5 Minuten (300 Sekunden) und der Bewertungszeiträume 1 ist, wird der Alarm am Ende von Minute 5 auf der Grundlage von Daten zwischen Minuten 1 und 5 ausgewertet. Am Ende von Minute 6 wird der Alarm dann auf der Grundlage der Daten aus den Minuten 2 bis 6 ausgewertet.

Wenn der Alarmzeitraum 10 Sekunden oder 30 Sekunden beträgt, wird der Alarm alle 10 Sekunden ausgewertet.

In der folgenden Abbildung ist die Alarmschwelle für einen Metrikalarm auf drei Einheiten festgelegt. Sowohl der Auswertungszeitraum als auch die Datenpunkte zum Alarm sind 3. Das heißt, wenn alle vorhandenen Datenpunkte in den letzten drei aufeinanderfolgenden Perioden über dem Schwellenwert liegen, wechselt der Alarm in den Status `ALARM`. In der Abbildung geschieht dies im dritten bis zum fünften Zeitraum. Bei Zeitraum 6 fällt der Wert unter den Schwellenwert, sodass einer der auszuwertenden Zeiträume keinen Schwellenwert verletzt, und der Alarmstatus ändert sich wieder in `OK`. Im neunten Zeitraum wird der Schwellenwert erneut verletzt, jedoch nur für einen Zeitraum. Daher bleibt der Alarmstatus `OK`.



Wenn Sie Evaluation Periods (Auswertungszeiträume) und Datapoints to Alarm (Datenpunkte zum Alarm) als unterschiedliche Werte konfigurieren, legen Sie einen „M von N“-Alarm fest. Datapoints to Alarm (Datenpunkte zum Alarm) ist („M“), Evaluation Periods (Auswertungszeiträume) ist („N“). Das Auswertungsintervall ist die Anzahl der Auswertungsperioden multipliziert mit der Zeitraumlänge. Wenn Sie beispielsweise 4 von 5 Datenpunkten mit einem Zeitraum von 1 Minute konfigurieren, beträgt das Auswertungsintervall 5 Minuten. Wenn Sie 3 von 3 Datenpunkten mit einem Zeitraum von 10 Minuten konfigurieren, beträgt das Auswertungsintervall 30 Minuten.

> **Note**
>
> Wenn kurz nach dem Erstellen eines Alarms Datenpunkte fehlen und die Metrik CloudWatch vor der Erstellung des Alarms gemeldet wurde, CloudWatch ruft es bei der Auswertung des Alarms die neuesten Datenpunkte ab, die vor der Erstellung des Alarms erstellt wurden.

Alarmaktionen

Sie können angeben, welche Aktionen ein Alarm ausführt, wenn er den Zustand zwischen den Zuständen OK, ALARM und INSUPFIZIENT_DATA ändert.

Die meisten Aktionen können für den Übergang in jeden der drei Zustände festgelegt werden. Mit Ausnahme der Auto-Scaling-Aktionen finden die Aktionen nur bei Zustandsübergängen statt und werden nicht erneut ausgeführt, wenn der Zustand über Stunden oder Tage anhält. Die Tatsache, dass für einen Alarm mehrere Aktionen zulässig sind, können Sie nutzen, um eine E-Mail zu senden, wenn ein Schwellenwert durchbrochen wird, und dann eine weitere, wenn die Durchbrechungsbedingung endet. So können Sie überprüfen, ob Ihre Skalierungs- oder Wiederherstellungsaktionen wie erwartet ausgelöst werden und wie gewünscht funktionieren.

Folgende Aktionen werden als Alarmaktionen unterstützt.

- Benachrichtigen Sie einen oder mehrere Subscriber mithilfe eines Themas von Amazon Simple Notification Service. Subscriber können sowohl Anwendungen als auch Personen sein. Mehr Informationen zu Amazon SNS finden Sie unter Was ist Amazon SNS?.
- Rufen Sie eine Lambda-Funktion auf. Dies ist die einfachste Methode für Sie, benutzerdefinierte Aktionen bei Änderungen des Alarmstatus zu automatisieren.
- Auf EC2-Metriken basierende Alarme können auch EC2-Aktionen ausführen, wie etwa das Anhalten, Beenden, Neustarten oder Wiederherstellen einer EC2-Instance. Weitere Informationen finden Sie unter Erstellen Sie Alarme, um eine EC2-Instance anzuhalten, zu beenden, neu zu starten oder wiederherzustellen.
- Alarme können auch Aktionen ausführen, um eine Auto-Scaling-Gruppe zu skalieren. Weitere Informationen finden Sie unter Schritte und Skalierungsrichtlinien für Amazon EC2 Auto Scaling.
- Alarme können OpsItems im Systems Manager Ops Center oder Vorfälle im AWS Systems Manager Incident Manager erstellt werden. Diese Aktionen werden nur ausgeführt, wenn der Alarm in den Zustand ALARM wechselt. Weitere Informationen finden Sie unter Konfiguration, CloudWatch um Alarme OpsItems aus Alarmen zu erstellen und Incident-Erstellung.

Lambda-Alarmaktionen

CloudWatch alarms garantiert einen asynchronen Aufruf der Lambda-Funktion für eine bestimmte Zustandsänderung, außer in den folgenden Fällen:

- Wenn die Funktion nicht existiert.

- When CloudWatch ist nicht berechtigt, die Lambda-Funktion aufzurufen.

Wenn der Lambda-Dienst nicht erreicht werden CloudWatch kann oder die Nachricht aus einem anderen Grund zurückgewiesen wird, versuchen Sie es CloudWatch erneut, bis der Aufruf erfolgreich ist. Lambda stellt die Nachricht in eine Warteschlange und verarbeitet Ausführungswiederholungen. Weitere Informationen zu diesem Ausführungsmodell, einschließlich Informationen darüber, wie Lambda mit Fehlern umgeht, finden Sie unter Asynchronous invocation im AWS Lambda Developer Guide.

Sie können eine Lambda-Funktion in demselben Konto oder in anderen AWS Konten aufrufen.

Wenn Sie einen Alarm angeben, um eine Lambda-Funktion als Alarmaktion aufzurufen, können Sie wählen, ob Sie den Funktionsnamen, den Funktionsalias oder eine bestimmte Version einer Funktion angeben möchten.

Wenn Sie eine Lambda-Funktion als Alarmaktion angeben, müssen Sie eine Ressourcenrichtlinie für die Funktion erstellen, damit der CloudWatch Dienstprinzipal die Funktion aufrufen kann.

Eine Möglichkeit, dies zu tun, ist die Verwendung von AWS CLI, wie im folgenden Beispiel:

```
aws lambda add-permission \
--function-name my-function-name \
--statement-id AlarmAction \
--action 'lambda:InvokeFunction' \
--principal lambda.alarms.cloudwatch.amazonaws.com \
--source-account 111122223333 \
--source-arn arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name
```

Alternativ können Sie eine Richtlinie erstellen, die einem der folgenden Beispiele ähnelt, und sie dann der Funktion zuweisen.

Im folgenden Beispiel wird das Konto angegeben, in dem sich der Alarm befindet, sodass nur Alarme in diesem Konto (111122223333) die Funktion aufrufen können.

```
{
    "Version": "2012-10-17",
    "Id": "default",
    "Statement": [{
        "Sid": "AlarmAction",
        "Effect": "Allow",
        "Principal": {
```

```
            "Service": "lambda.alarms.cloudwatch.amazonaws.com"
        },
        "Action": "lambda:InvokeFunction",
        "Resource": "arn:aws:lambda:us-east-1:444455556666:function:function-name",
        "Condition": {
            "StringEquals": {
                "AWS:SourceAccount": "111122223333"
            }
        }
    }]
}
```

Das folgende Beispiel hat einen engeren Gültigkeitsbereich, sodass nur der angegebene Alarm im angegebenen Konto die Funktion aufrufen kann.

```
{
  "Version": "2012-10-17",
  "Id": "default",
  "Statement": [
    {
      "Sid": "AlarmAction",
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.alarms.cloudwatch.amazonaws.com"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:us-east-1:444455556666:function:function-name",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "111122223333",
          "AWS:SourceArn": "arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name"
        }
      }
    }]
}
```

Es wird nicht empfohlen, eine Richtlinie zu erstellen, in der kein Quellkonto angegeben ist, da solche Richtlinien anfällig für „Probleme durch verwirrte Stellvertreter“ sind.

Ereignisobjekt, das von an CloudWatch Lambda gesendet wurde

Wenn Sie eine Lambda-Funktion als Alarmaktion konfigurieren, CloudWatch übermittelt sie eine JSON-Nutzlast an die Lambda-Funktion, wenn sie die Funktion aufruft. Diese JSON-Nutzlast dient als

Ereignisobjekt für die Funktion. Sie können Daten aus diesem JSON-Objekt extrahieren und in Ihrer Funktion verwenden. Das folgende ist ein Beispiel eines Ereignisobjekts aus einem Metrikalarm.


```
{
  'source': 'aws.cloudwatch',
  'alarmArn': 'arn:aws:cloudwatch:us-east-1:444455556666:alarm:lambda-demo-metric-
alarm',
  'accountId': '444455556666',
  'time': '2023-08-04T12:36:15.490+0000',
  'region': 'us-east-1',
  'alarmData': {
    'alarmName': 'lambda-demo-metric-alarm',
    'state': {
      'value': 'ALARM',
      'reason': 'test',
      'timestamp': '2023-08-04T12:36:15.490+0000'
    },
    'previousState': {
      'value': 'INSUFFICIENT_DATA',
      'reason': 'Insufficient Data: 5 datapoints were unknown.',
      'reasonData':
 '{"version":"1.0","queryDate":"2023-08-04T12:31:29.591+0000","statistic":"Average","period":60
[],"threshold":5.0,"evaluatedDatapoints":[{"timestamp":"2023-08-04T12:30:00.000+0000"},
{"timestamp":"2023-08-04T12:29:00.000+0000"},
{"timestamp":"2023-08-04T12:28:00.000+0000"},
{"timestamp":"2023-08-04T12:27:00.000+0000"},
{"timestamp":"2023-08-04T12:26:00.000+0000"}]}',
      'timestamp': '2023-08-04T12:31:29.595+0000'
    },
    'configuration': {
      'description': 'Metric Alarm to test Lambda actions',
      'metrics': [
        {
          'id': '1234e046-06f0-a3da-9534-EXAMPLEe4c',
          'metricStat': {
            'metric': {
              'namespace': 'AWS/Logs',
              'name': 'CallCount',
              'dimensions': {
                'InstanceId': 'i-12345678'
              }
            },
            'period': 60,
```

```
            'stat': 'Average',
            'unit': 'Percent'
          },
          'returnData': True
        }
      ]
    }
  }
}
```

Das folgende ist ein Beispiel eines Ereignisobjekts aus einem zusammengesetzten Alarm.

```
{
  'source': 'aws.cloudwatch',
  'alarmArn': 'arn:aws:cloudwatch:us-east-1:111122223333:alarm:SuppressionDemo.Main',
  'accountId': '111122223333',
  'time': '2023-08-04T12:56:46.138+0000',
  'region': 'us-east-1',
  'alarmData': {
    'alarmName': 'CompositeDemo.Main',
    'state': {
      'value': 'ALARM',
      'reason': 'arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild transitioned to ALARM at Friday 04
 August, 2023 12:54:46 UTC',
      'reasonData': '{"triggeringAlarms":[{"arn":"arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild","state":
{"value":"ALARM","timestamp":"2023-08-04T12:54:46.138+0000"}}]}',
      'timestamp': '2023-08-04T12:56:46.138+0000'
    },
    'previousState': {
      'value': 'ALARM',
      'reason': 'arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild transitioned to ALARM at Friday 04
 August, 2023 12:54:46 UTC',
      'reasonData': '{"triggeringAlarms":[{"arn":"arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild","state":
{"value":"ALARM","timestamp":"2023-08-04T12:54:46.138+0000"}}]}',
      'timestamp': '2023-08-04T12:54:46.138+0000',
      'actionsSuppressedBy': 'WaitPeriod',
      'actionsSuppressedReason': 'Actions suppressed by WaitPeriod'
    },
    'configuration': {
```

```
      'alarmRule': 'ALARM(CompositeDemo.FirstChild) OR
 ALARM(CompositeDemo.SecondChild)',
      'actionsSuppressor': 'CompositeDemo.ActionsSuppressor',
      'actionsSuppressorWaitPeriod': 120,
      'actionsSuppressorExtensionPeriod': 180
    }
  }
}
```

Konfiguration, wie Alarme fehlende Daten behandeln CloudWatch

Manchmal wird nicht jeder erwartete Datenpunkt für eine Metrik gemeldet CloudWatch. Dies kann z. B. der Fall sein, wenn eine Verbindung unterbrochen wird, ein Server ausfällt oder wenn eine Metrik Daten grundsätzlich nur intermittierend meldet.

CloudWatch ermöglicht es Ihnen, festzulegen, wie fehlende Datenpunkte bei der Auswertung eines Alarms behandelt werden sollen. Dies hilft Ihnen, Ihren Alarm so zu konfigurieren, dass er nur dann in den `ALARM`-Zustand wechselt, wenn dies für den überwachten Datentyp angemessen ist. Sie können Fehlalarme vermeiden, wenn fehlende Daten kein Problem darstellen.

Ähnlich wie sich jeder Alarm immer in einem von drei Zuständen befindet, CloudWatch fällt jeder spezifische Datenpunkt, an den gemeldet wird, in eine von drei Kategorien:

- Keine Verletzung (innerhalb des Schwellenwerts)
- Verletzung (verletzt den Schwellenwert)
- Fehlen

Für jeden Alarm können Sie angeben CloudWatch , dass fehlende Datenpunkte wie folgt behandelt werden sollen:

- `notBreaching` – Fehlende Datenpunkte werden als „gut“ und innerhalb der Schwelle liegend behandelt
- `breaching` – Fehlende Datenpunkte werden als „ungültig“ und außerhalb der Schwelle liegend behandelt
- `ignore` – Der aktuelle Alarmstatus wird beibehalten
- `missing` – Wenn alle Datenpunkte im Alarmauswertungsbereich fehlen, wechselt der Alarm zu INSUPFIZIENT_DATA.

Die beste Wahl hängt von der Art der Metrik und dem Zweck des Alarms ab. Wenn Sie beispielsweise einen Anwendungs-Rollback-Alarm mithilfe einer Metrik erstellen, die kontinuierlich Daten meldet, sollten Sie fehlende Datenpunkte möglicherweise als Sicherheitsverletzung behandeln, da dies darauf hindeuten könnte, dass etwas nicht stimmt. Aber für eine Metrik, die nur Datenpunkte generiert, wenn ein Fehler auftritt, z. B. `ThrottledRequests` in Amazon DynamoDB, sollten Sie fehlende Daten als `notBreaching` behandeln. Das Standardverhalten ist `missing`.

> **Important**
>
> Für Amazon EC2-Metriken konfigurierte Alarme können vorübergehend den Status INSUFFICIENT_DATA annehmen, wenn Metrikdatenpunkte fehlen. Dies ist selten, kann aber passieren, wenn die Metrikberichterstattung unterbrochen wird, selbst wenn die Amazon EC2 EC2-Instance fehlerfrei ist. Für Alarme auf Amazon EC2-Metriken, die so konfiguriert sind, dass sie Stopp-, Beenden-, Neustart- oder Wiederherstellungsaktionen ausführen, empfehlen wir, diese Alarme so zu konfigurieren, dass fehlende Daten als solche behandelt werden und dass diese Alarme nur ausgelöst werden`missing`, wenn sie sich im ALARM-Status befinden.

Wenn Sie die beste Option für Ihren Alarm wählen, werden unnötige und irreführende Änderungen an Alarmbedingungen verhindert und der Zustand Ihres Systems wird genauer angeben.

> **Important**
>
> Alarme, die Metriken im `AWS/DynamoDB`-Namespace bewerten, ignorieren immer fehlende Daten, auch wenn Sie eine andere Option wählen, wie der Alarm fehlende Daten behandeln soll. Wenn für eine `AWS/DynamoDB`-Metrik Daten fehlen, verbleiben Alarme, die diese Metrik auswerten, in ihrem aktuellen Zustand.

Wie der Alarmstatus bei fehlenden Daten ausgewertet wird

Immer wenn ein Alarm auswertet, ob der Status geändert werden soll, CloudWatch versucht er, eine höhere Anzahl von Datenpunkten als die als Evaluierungszeiträume angegebene Anzahl abzurufen. Die genaue Anzahl der Datenpunkte, die er abzurufen versucht, hängt von der Länge des Alarmzeitraums und davon ab, ob er auf einer Metrik mit Standardauflösung oder hoher Auflösung basiert. Der Zeitrahmen der Datenpunkte, die sie abzurufen versucht, ist der Auswertungsbereich.

Sobald diese Datenpunkte CloudWatch abgerufen wurden, passiert Folgendes:

- Wenn keine Datenpunkte im Bewertungsbereich fehlen, wird der Alarm auf der Grundlage der zuletzt erfassten Datenpunkte CloudWatch ausgewertet. Die Anzahl der ausgewerteten Datenpunkte entspricht den Auswertungszeiträumen für den Alarm. Die zusätzlichen Datenpunkte von weiter hinten im Auswertungsbereich werden nicht benötigt und ignoriert.
- Wenn einige Datenpunkte im Bewertungsbereich fehlen, aber die Gesamtzahl der vorhandenen Datenpunkte, die erfolgreich aus dem Bewertungsbereich abgerufen wurden, gleich oder größer als die Bewertungszeiträume des Alarms ist, CloudWatch bewertet der Alarmstatus auf der Grundlage der letzten erfolgreich abgerufenen realen Datenpunkte, einschließlich der erforderlichen zusätzlichen Datenpunkte, die weiter hinten im Bewertungsbereich liegen. In diesem Fall wird der von Ihnen eingestellte Wert für die Behandlung fehlender Daten nicht benötigt und ignoriert.
- Wenn einige Datenpunkte im Bewertungsbereich fehlen und die Anzahl der tatsächlich abgerufenen Datenpunkte niedriger ist als die Anzahl der Evaluierungsperioden des Alarms, CloudWatch füllt die fehlenden Datenpunkte mit dem Ergebnis auf, das Sie für die Behandlung fehlender Daten angegeben haben, und wertet dann den Alarm aus. Es werden jedoch alle realen Datenpunkte im Bewertungsbereich in die Auswertung einbezogen. CloudWatch verwendet fehlende Datenpunkte nur so selten wie möglich.

> **Note**
>
> Ein besonderer Fall dieses Verhaltens besteht darin, dass CloudWatch Alarme den letzten Satz von Datenpunkten für einen bestimmten Zeitraum wiederholt neu auswerten, nachdem die Metrik aufgehört hat zu fließen. Diese Neuauswertung kann dazu führen, dass der Alarm den Status ändert und Aktionen erneut ausführt, wenn er den Status unmittelbar vor dem Stoppen des Messdatenstroms geändert hatte. Um dieses Verhalten zu verhindern, verwenden Sie kürzere Zeiträume.

Die folgenden Tabellen zeigen Beispiele für das Verhalten der Alarmauswertung. In der ersten Tabelle haben die Datenpunkte für Alarm - und Bewertungszeiträume jeweils den Wert 3. CloudWatch ruft bei der Auswertung des Alarms die 5 neuesten Datenpunkte ab, falls einige der letzten 3 Datenpunkte fehlen. 5 ist der Bewertungsbereich für den Alarm.

In Spalte 1 werden die fünf letzten Datenpunkte angezeigt, da der Auswertungsbereich 5 beträgt. Diese Datenpunkte werden mit dem letzten Datenpunkt auf der rechten Seite angezeigt. 0 ist ein nicht überschreitender Datenpunkt, X ein überschreitender Datenpunkt und - ein fehlender Datenpunkt.

In Spalte 2 ist angegeben, wie viele der 3 erforderlichen Datenpunkte fehlen. Obwohl die 5 zuletzt hinzugekommenen Datenpunkte ausgewertet werden, sind zur Bewertung des Alarmstatus nur 3 davon nötig (gemäß der Einstellung für Evaluation Periods (Auswertungszeiträume)). Die Anzahl der Datenpunkte in Spalte 2 ist die Anzahl der Datenpunkte, die ausgefüllt sein muss. Dabei wird die Einstellung zur Behandlung fehlender Daten verwendet.

In den Spalten 3-6 sind die Spaltenüberschriften die möglichen Werte für die Behandlung fehlender Daten. Die Zeilen in diesen Spalten zeigen den Alarmstatus an, der für jede dieser möglichen Methoden zur Behandlung fehlender Daten festgelegt ist.

Datenpunkte	Anzahl der Datenpunkte, die ausgefüllt werden müssen	MISSING	IGNORE	ÜBERSCHREITEND	NICHT ÜBERSCHREITEND
0 - X - X	0	OK	OK	OK	OK
- - - - 0	2	OK	OK	OK	OK
- - - - -	3	INSUFFICIENT_DATA	Aktuellen Status beibehalten	ALARM	OK
0 X X - X	0	ALARM	ALARM	ALARM	ALARM
- - X - -	2	ALARM	Aktuellen Status beibehalten	ALARM	OK

In der zweiten Zeile der vorhergehenden Tabelle bleibt der Alarm OK, auch wenn fehlende Daten als Überschreitung behandelt werden, weil der eine vorhandene Datenpunkt nicht überschreitend ist. Dies wird zusammen mit zwei fehlenden Datenpunkten ausgewertet, die als Überschreitung behandelt werden. Wenn dieser Alarm das nächste Mal ausgewertet wird, werden die Daten, die noch fehlen, auf ALARM gesetzt, da dieser nicht verletzende Datenpunkt nicht mehr im Auswertebereich liegt.

Die dritte Zeile, in der alle fünf letzten Datenpunkte fehlen, veranschaulicht, wie sich die verschiedenen Einstellungen für die Behandlung fehlender Daten auf den Alarmzustand auswirken. Wenn fehlende Datenpunkte als Verletzung betrachtet werden, geht der Alarm in den ALARM-Status, während, wenn sie als nicht verletzt betrachtet werden, dann geht der Alarm in den OK-Zustand über. Wenn fehlende Datenpunkte ignoriert werden, behält der Alarm den aktuellen Zustand vor den fehlenden Datenpunkten bei. Und wenn fehlende Datenpunkte nur als fehlend angesehen werden, dann hat der Alarm nicht genügend aktuelle reale Daten, um eine Auswertung durchzuführen und geht in den Zustand INSUFFIZIENT_DATA.

In der vierten Zeile geht der Alarm in allen Fällen in den Zustand `ALARM` über, da die drei jüngsten Datenpunkte verletzt werden und die Auswertungszeiträume und Datenpunkte zum Alarm des Alarms beide auf 3 gesetzt sind. In diesem Fall wird der fehlende Datenpunkt ignoriert und die Einstellung für die Auswertung fehlender Daten ist nicht erforderlich, da drei reale Datenpunkte ausgewertet werden müssen.

Zeile 5 stellt einen Sonderfall der Alarmauswertung dar, der als vorzeitiger Alarmzustand bezeichnet wird. Weitere Informationen finden Sie unter Vermeidung vorzeitiger Übergänge in den Alarmzustand.

In der nächsten Tabelle ist Period (Zeitraum) erneut auf 5 Minuten gesetzt, und Datapoints to Alarm (Datenpunkte zum Alarm) ist nur 2, während Evaluation Periods (Auswertungszeiträume) 3 ist. Die ist ein 2-aus-3, M-aus-N-Alarm.

Der Auswertungsbereich beträgt 5. Dies ist die maximale Anzahl der zuletzt abgerufenen Datenpunkte und kann verwendet werden, falls einige Datenpunkte fehlen.

Datenpunkte	Anzahl fehlender Datenpunkte	FEHLEND	IGNORE	ÜBERSCHREITEND	NICHT ÜBERSCHREITEND
0 - X - X	0	`ALARM`	`ALARM`	`ALARM`	`ALARM`
0 0 X 0 X	0	`ALARM`	`ALARM`	`ALARM`	`ALARM`
0 - X - -	1	`OK`	`OK`	`ALARM`	`OK`
- - - - 0	2	`OK`	`OK`	`ALARM`	`OK`

Datenpunkte	Anzahl fehlender Datenpunkte	FEHLEND	IGNORE	ÜBERSCHREITEND	NICHT ÜBERSCHREITEND
- - - - X	2	ALARM	Aktuellen Status beibehalten	ALARM	OK

In den Zeilen 1 und 2 geht der Alarm immer in den ALARM-Zustand, da 2 der 3 letzten Datenpunkte verletzt werden. In Zeile 2 werden die beiden ältesten Datenpunkte im Auswertebereich nicht benötigt, da keiner der 3 jüngsten Datenpunkte fehlt, daher werden diese beiden älteren Datenpunkte ignoriert.

In den Zeilen 3 und 4 geht der Alarm nur dann in den ALARM-Zustand, wenn fehlende Daten als Verletzung behandelt werden. In diesem Fall werden die beiden letzten fehlenden Datenpunkte beide als Verletzung behandelt. In Zeile 4 stellen diese beiden fehlenden Datenpunkte, die als Verletzung behandelt werden, die zwei notwendigen Datenpunkte bereit, um den ALARM-Zustand auszulösen.

Zeile 5 stellt einen Sonderfall der Alarmauswertung dar, der als vorzeitiger Alarmzustand bezeichnet wird. Weitere Informationen finden Sie im folgenden Abschnitt.

Vermeidung vorzeitiger Übergänge in den Alarmzustand

CloudWatch Die Auswertung von Alarmen beinhaltet Logik zur Vermeidung von Fehlalarmen, bei denen der Alarm vorzeitig in den ALARM-Zustand übergeht, wenn die Daten unterbrochen werden. Das Beispiel, das in Zeile 5 in den Tabellen im vorherigen Abschnitt gezeigt wird, veranschaulicht diese Logik. In diesen Zeilen und in den folgenden Beispielen beträgt der Auswertungszeitraum 3 und der Auswertungsbereich 5 Datenpunkte. Datenpunkte zum Alarm sind 3, mit Ausnahme des Beispiels M von N, wo Datenpunkte zum Alarm 2 sind.

Angenommen, die neuesten Daten eines Alarms sind - - - - X, mit vier fehlenden Datenpunkten und dann einem Datenpunkt, der verletzt wird, als neuestem Datenpunkt. Da der nächste Datenpunkt möglicherweise nicht verletzt wird, geht der Alarm nicht sofort in den ALARM-Zustand, wenn die Daten entweder - - - - X oder - - - X - sind und Datenpunkte zum Alarm 3 sind. Auf diese Weise werden Fehlalarme vermieden, wenn der nächste Datenpunkt nicht verletzt wird und dazu führt, dass die Daten - - - X 0 oder - - X - 0 sind.

Wenn jedoch die letzten paar Datenpunkte - - X - - sind, geht der Alarm in den ALARM-Zustand, auch wenn fehlende Datenpunkte als fehlend behandelt werden. Dies liegt daran, dass Alarme immer in den ALARM-Zustand gehen, wenn der älteste verfügbare verletzende Datenpunkt in der Anzahl der Datenpunkte während der Auswertungszeiträume mindestens so alt ist wie der Wert von Mit Alarm zu versehende Datenpunkte und alle anderen neueren Datenpunkte eine Verletzung darstellen oder fehlen. In diesem Fall geht der Alarm auch dann in den ALARM-Zustand, wenn die Gesamtzahl der verfügbaren Datenpunkte kleiner als M (Datenpunkte zum Alarm) ist.

Diese Alarmlogik gilt auch für „M von N“ Alarme. Wenn der älteste verletzte Datenpunkt während des Auswertungsbereichs mindestens so alt ist wie der Wert von Datenpunkte zum Alarm und alle neueren Datenpunkte entweder verletzt oder fehlen, geht der Alarm unabhängig vom Wert von M (Datenpunkte zum Alarm).

Hochauflösende Alarme

Wenn Sie einen Alarm für eine hochauflösende Metrik festlegen, können Sie einen hochauflösenden Alarm für einen Zeitraum von 10 Sekunden oder 30 Sekunden oder einen regelmäßigen Alarm für einen Zeitraum festlegen, der ein Mehrfaches von 60 Sekunden beträgt. Für hochauflösende Alarme ist eine höhere Gebühr zu zahlen. Weitere Informationen zu hochauflösenden Metriken finden Sie unter Veröffentlichen von benutzerdefinierten -Metriken.

Alarme bei mathematischen Ausdrücken

Sie können einen Alarm für das Ergebnis eines mathematischen Ausdrucks einrichten, der auf einer oder mehreren Metriken basiert. CloudWatch Ein mathematischer Ausdruck, der für einen Alarm verwendet wird, kann bis zu 10 Metriken umfassen. Jede Metrik muss den gleichen Zeitraum verwenden.

Bei einem Alarm, der auf einem mathematischen Ausdruck basiert, können Sie angeben, wie Sie mit fehlenden Datenpunkten umgehen CloudWatch möchten. In diesem Fall wird der Datenpunkt als fehlend betrachtet, wenn der mathematische Ausdruck keinen Wert für diesen Datenpunkt liefert.

Alarme, die auf mathematischen Ausdrücken basieren, können keine Amazon-EC2-Aktionen ausführen.

Weitere Informationen über metrische mathematische Ausdrücke und Syntax finden Sie unter Verwenden von Metrikberechnungen.

Auf Perzentilen basierende CloudWatch Alarme und Stichproben mit niedrigen Datenmengen

Wenn Sie ein Perzentil als Statistik für einen Alarm festlegen, können Sie angeben, was zu tun ist, wenn nicht genügend Daten für eine gute statistische Bewertung vorliegen. Sie können festlegen, dass der Alarm trotzdem statistisch bewertet wird und der Alarmstatus möglicherweise geändert wird. Alternativ können Sie festlegen, dass der Alarm die Metrik bei einer kleinen Stichprobe ignoriert und mit der Bewertung wartet, bis ausreichend Daten vorliegen, um statistische Signifikanz zu erzielen.

Für Perzentile zwischen0,5 (inklusive) und 1,00 (exklusive) wird diese Einstellung verwendet, wenn weniger als 10/(1-Perzentil) Datenpunkte im Bewertungszeitraum vorhanden sind. Diese Einstellung würde beispielsweise verwendet werden, wenn weniger als 1 000 Beispiele für einen Alarm auf einem p99 Perzentil vorhanden sind. Für Perzentile zwischen 0 und 0,5 (exklusive) wird die Einstellung verwendet, wenn weniger als 10/Perzentil Datenpunkte vorhanden sind.

Gemeinsame Merkmale von Alarmen CloudWatch

Die folgenden Funktionen gelten für alle CloudWatch Alarme:

- Die Anzahl der Alarme, die Sie erstellen können, ist unbegrenzt. Um einen Alarm zu erstellen oder zu aktualisieren, verwenden Sie die CloudWatch Konsole, die PutMetricAlarmAPI-Aktion oder den put-metric-alarmBefehl in AWS CLI.
- Alarmnamen dürfen nur UTF-8-Zeichen und keine ASCII-Kontrolleingabezeichen enthalten
- Sie können einige oder alle aktuell konfigurierten Alarme und alle Alarme in einem bestimmten Status auflisten, indem Sie die CloudWatch Konsole, die DescribeAlarmsAPI-Aktion oder den Befehl describe-alarms in der verwenden. AWS CLI
- Sie können Alarme deaktivieren und aktivieren, indem Sie die EnableAlarmActionsAPI-Aktionen DisableAlarmActionsund oder die enable-alarm-actionsBefehle disable-alarm-actionsund in der verwenden. AWS CLI
- Sie können einen Alarm testen, indem Sie ihn mit der SetAlarmStateAPI-Aktion oder dem set-alarm-stateBefehl im auf einen beliebigen Status setzen AWS CLI. Diese temporäre Statusänderung dauert nur bis zum nächsten Alarmvergleich.
- Sie können einen Alarm für eine benutzerdefinierte Metrik erstellen, bevor Sie diese benutzerdefinierte Metrik selbst erstellen. Damit der Alarm gültig ist, müssen Sie alle Dimensionen für die benutzerdefinierte Metrik zusätzlich zum Namen des Namespace und der Metrik sowie der

Alarmdefinition einfügen. Dazu können Sie die PutMetricAlarmAPI-Aktion oder den put-metric-alarmBefehl in der verwenden AWS CLI.

- Sie können den Verlauf eines Alarms mithilfe der CloudWatch Konsole, der DescribeAlarmHistoryAPI-Aktion oder des describe-alarm-historyBefehls in der anzeigen AWS CLI. CloudWatch speichert den Alarmverlauf für 30 Tage. Jeder Statusübergang wird mit einem eindeutigen Zeitstempel versehen. In seltenen Fällen kann es vorkommen, dass der Verlauf mehr als eine Benachrichtigung für eine Statusänderung anzeigt. Mit dem Zeitstempel können Sie eindeutige Änderungen des Status bestätigen.
- Sie können Alarme über die Option Favoriten und zuletzt verwendete Alarme im Navigationsbereich der CloudWatch Konsole zu Favoriten hinzufügen, indem Sie den Mauszeiger über den Alarm bewegen, den Sie als Favorit markieren möchten, und das Sternsymbol neben dem Alarm auswählen.
- Die Anzahl der Auswertungszeiträume für einen Alarm multipliziert mit der Länge der einzelnen Auswertungszeiträume darf einen Tag nicht überschreiten.

> **Note**
>
> Einige AWS Ressourcen senden CloudWatch unter bestimmten Bedingungen keine Metrikdaten an.
> Beispielsweise sendet Amazon EBS möglicherweise keine Messdaten für ein verfügbares Volume, das nicht an eine Amazon-EC2-Instance angehängt ist, da für dieses Volume keine Messdatenaktivität überwacht werden muss. Wenn Sie einen Alarm für eine solche Metrik festgelegt haben, werden Sie möglicherweise feststellen, dass sich ihr Status auf `INSUFFICIENT_DATA` ändert. Dies kann darauf hindeuten, dass Ihre Ressource inaktiv ist, und bedeutet nicht unbedingt, dass es ein Problem gibt. Sie können festlegen, wie jeder Alarm fehlende Daten behandelt. Weitere Informationen finden Sie unter Konfiguration, wie Alarme fehlende Daten behandeln CloudWatch .

Bewährte Alarmempfehlungen für AWS Dienste

CloudWatch bietet Empfehlungen für out-of-the Box-Alarme. Es wird empfohlen, diese CloudWatch Alarme für Metriken zu erstellen, die von anderen AWS Diensten veröffentlicht werden. Mit diesen Empfehlungen können Sie die Metriken ermitteln, für die ein Alarm eingestellt werden sollte, um bewährte Methoden zur Überwachung zu befolgen. In den Empfehlungen werden auch die

einzustellenden Alarmschwellenwerte vorgeschlagen. Wenn Sie diese Empfehlungen befolgen, können Sie wichtige Überwachungen Ihrer AWS Infrastruktur nicht verpassen.

Um die Alarmempfehlungen zu finden, verwenden Sie den Metrikbereich der CloudWatch Konsole und wählen den Filter für Alarmempfehlungen aus. Wenn Sie in der Konsole zu den empfohlenen Alarmen navigieren und dann einen empfohlenen Alarm erstellen, CloudWatch können Sie einige der Alarmeinstellungen vorab ausfüllen. Bei einigen empfohlenen Alarmen ist der Alarmschwellenwert ebenfalls vorausgefüllt. Sie können die Konsole auch verwenden, um infrastructure-as-code Alarmdefinitionen für empfohlene Alarme herunterzuladen und dann diesen Code verwenden, um den Alarm in AWS CloudFormation AWS CLI, der oder Terraform zu erstellen.

Die Liste der empfohlenen Alarme finden Sie auch unter Empfohlene Alarme.

Die von Ihnen erstellten Alarme werden Ihnen genauso berechnet wie für alle anderen Alarme, die Sie in erstellen. CloudWatch Für die Nutzung der Empfehlungen fallen keine zusätzlichen Gebühren an. Weitere Informationen finden Sie unter CloudWatch Amazon-Preise.

Finden und erstellen von empfohlenen Alarmen

Gehen Sie wie folgt vor, um die Metriken zu finden, für die das Einrichten von Alarmen CloudWatch empfohlen wird, und optional, um einen dieser Alarme zu erstellen. Das erste Verfahren erklärt, wie Sie die Metriken finden, die über empfohlene Alarme verfügen, und wie Sie einen dieser Alarme erstellen.

Sie können auch einen Massen-Download von infrastructure-as-code Alarmdefinitionen für alle empfohlenen Alarme in einem AWS Namespace, z. B. `AWS/Lambda` oder`AWS/S3`, abrufen. Diese Anweisungen finden Sie an späterer Stelle in diesem Thema.

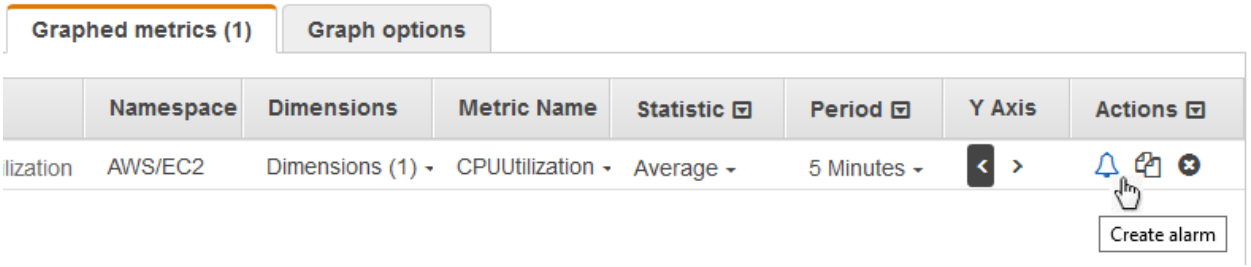
So finden Sie die Metriken mit empfohlenen Alarmen und erstellen einen einzelnen empfohlenen Alarm

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken) All metrics (Alle Metriken) aus.
3. Wählen Sie oberhalb der Metriktabelle die Option Alarmempfehlungen aus.

   Die Liste der Metrik-Namespaces ist so gefiltert, dass sie nur die Metriken enthält, die Alarmempfehlungen enthalten und die Services in Ihrem Konto veröffentlichen.
4. Wählen Sie den Namespace für einen Service aus.

Die Liste der Metriken unter diesem Namespace wird so gefiltert, dass sie nur solche enthält, die Alarmempfehlungen enthalten.

5. Um die Alarmabsicht und den empfohlenen Schwellenwert für eine Metrik zu sehen, wählen Sie Details anzeigen.
6. Um einen Alarm für eine der Metriken zu öffnen, führen Sie einen der folgenden Schritte aus:
   - Um den Alarm über die Konsole zu erstellen, gehen Sie wie folgt vor:
     a. Aktivieren Sie das Kontrollkästchen für die Metrik und wählen Sie die Registerkarte Grafische Metriken aus.
     b. Wählen Sie das Alarmsymbol aus.



     Der Assistent zur Erstellung von Alarmen wird angezeigt, wobei der Name der Metrik, die Statistik und der Zeitraum auf der Grundlage der Alarmempfehlung ausgefüllt werden. Wenn die Empfehlung einen bestimmten Schwellenwert enthält, ist dieser Wert ebenfalls vorausgefüllt.

     c. Wählen Sie Weiter aus.
     d. Wählen Sie unter Benachrichtigung ein SNS-Thema aus, das benachrichtigt werden soll, wenn sich der Alarm im Status `ALARM`, `OK` oder `INSUFFICIENT_DATA` befindet.

        Um zu erreichen, dass der Alarm mehrere Benachrichtigungen für den gleichen Alarmstatus oder für verschiedene Statuswerte sendet, wählen Sie Benachrichtigung hinzufügen.

        Damit der Alarm keine Benachrichtigungen sendet, wählen Sie Remove (Entfernen).
     e. Um den Alarm Auto-Scaling- oder EC2-Aktionen durchführen zu lassen, wählen Sie die entsprechende Schaltfläche und wählen Sie den Alarmstatus und die auszuführende Aktion.
     f. Wenn Sie fertig sind, wählen Sie Weiter.

g. Geben Sie einen Namen und eine Beschreibung für den Alarm ein. Der Name darf nur ASCII-Zeichen enthalten. Wählen Sie anschließend Weiter.

h. Bestätigen Sie unter Preview and create (Vorschau und erstellen), dass die Informationen und Bedingungen den Anforderungen entsprechen, und wählen Sie dann Create alarm (Alarm erstellen).

- Um eine infrastructure-as-code Alarmdefinition zur Verwendung in AWS CloudFormation, AWS CLI, oder Terraform herunterzuladen, wählen Sie Alarmcode herunterladen und wählen Sie das gewünschte Format aus. Der heruntergeladene Code wird die empfohlenen Einstellungen für den Metriknamen, die Statistik und den Schwellenwert enthalten.

Um infrastructure-as-code Alarmdefinitionen für alle empfohlenen Alarme für einen Dienst herunterzuladen AWS

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken) All metrics (Alle Metriken) aus.
3. Wählen Sie oberhalb der Metriktabelle die Option Alarmempfehlungen aus.

   Die Liste der Metrik-Namespaces ist so gefiltert, dass sie nur die Metriken enthält, die Alarmempfehlungen enthalten und die Services in Ihrem Konto veröffentlichen.
4. Wählen Sie den Namespace für einen Service aus.

   Die Liste der Metriken unter diesem Namespace wird so gefiltert, dass sie nur solche enthält, die Alarmempfehlungen enthalten.
5. Der Download-Alarmcode zeigt an, wie viele Alarme für die Metriken in diesem Namespace empfohlen werden. Um infrastructure-as-code Alarmdefinitionen für alle empfohlenen Alarme herunterzuladen, wählen Sie Alarmcode herunterladen und wählen Sie dann das gewünschte Codeformat aus.

Empfohlene Alarme

In den folgenden Abschnitten sind die Metriken aufgeführt, für die Sie Alarme nach bewährten Methoden einrichten sollten. Für jede Metrik werden auch die Dimensionen, die Alarmabsicht, der empfohlene Schwellenwert, die Schwellenwertbegründung sowie die Länge des Zeitraums und die Anzahl der Datenpunkte angezeigt.

Einige Metriken werden möglicherweise zweimal in der Liste angezeigt. Dies ist der Fall, wenn unterschiedliche Alarme für verschiedene Kombinationen von Dimensionen dieser Metrik empfohlen werden.

Datenpunkte bis Alarm ist die Anzahl der Datenpunkte, die überschritten werden müssen, um den Alarm in den ALARM-Status zu versetzen. Die Auswertungszeiträume geben die Anzahl der Zeiträume an, die bei der Auswertung des Alarms berücksichtigt werden. Wenn diese Zahlen identisch sind, geht der Alarm nur dann in den ALARM-Status über, wenn diese Anzahl aufeinanderfolgender Zeiträume Werte aufweist, die den Schwellenwert überschreiten. Wenn Datenpunkte bis Alarm niedriger ist als die Auswertungszeiträume, dann handelt es sich um einen „M aus N“-Alarm und der Alarm geht in den ALARM-Zustand über, wenn zumindest die Datenpunkte für Datenpunkte bis zum Alarm innerhalb eines beliebigen Satzes von Datenpunkten der Auswertungszeiträume überschritten werden. Weitere Informationen finden Sie unter Auswerten eines Alarms.

Themen

- Amazon API Gateway
- Amazon EC2 Auto Scaling
- Amazon CloudFront
- Amazon Cognito
- Amazon DynamoDB
- Amazon EBS
- Amazon EC2
- Amazon ElastiCache
- Amazon EC2 (AWS/ElasticGPUs)
- Amazon ECS
- Amazon ECS mit Container Insights
- Amazon EFS
- Amazon EKS mit Container Insights
- Amazon Kinesis Data Streams
- Lambda
- Lambda Insights
- Amazon VPC (AWS/NATGateway)
- AWS Privater Link (AWS/PrivateLinkEndpoints)

- AWS Privater Link (AWS/PrivateLinkServices)
- Amazon RDS
- Amazon Route 53 Public Data Plane
- Amazon S3
- S3ObjectLambda
- Amazon SNS
- Amazon SQS
- AWS VPN

Amazon API Gateway

4XXError

Abmessungen: ApiName Bühne

Alarmbeschreibung: Dieser Alarm erkennt eine hohe Rate von clientseitigen Fehlern. Dies kann auf ein Problem mit den Autorisierungs- oder Client-Anfrageparametern hinweisen. Es könnte auch bedeuten, dass eine Ressource entfernt wurde oder dass ein Client eine Ressource anfordert, die nicht existiert. Erwägen Sie, CloudWatch Logs zu aktivieren und nach Fehlern zu suchen, die die 4XX-Fehler verursachen könnten. Erwägen Sie außerdem, detaillierte CloudWatch Metriken zu aktivieren, um diese Metrik pro Ressource und Methode anzuzeigen und die Fehlerquelle einzugrenzen. Fehler können auch durch eine Überschreitung des konfigurierten Drosselungslimits verursacht werden. Wenn in den Antworten und Protokollen eine hohe und unerwartete Anzahl von 429-Fehlern gemeldet wird, folgen Sie dieser Anleitung, um dieses Problem zu beheben.

Absicht: Dieser Alarm kann eine hohe Anzahl von clientseitigen Fehlern bei den API-Gateway-Anfragen erkennen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 0,05

Begründung des Schwellenwerts: Mit dem vorgeschlagenen Schwellenwert wird erkannt, wenn bei mehr als 5 % der gesamten Anfragen 4XX-Fehler auftreten. Sie können den Schwellenwert jedoch so einstellen, dass er dem Datenverkehr der Anfragen sowie den akzeptablen Fehlerraten entspricht. Sie können auch historische Daten analysieren, um die akzeptable Fehlerrate für den

Anwendungs-Workload zu bestimmen, und den Schwellenwert dann entsprechend anpassen. Bei häufig auftretenden 4XX-Fehlern muss ein Alarm ausgelöst werden. Die Einstellung eines sehr niedrigen Schwellenwerts kann jedoch dazu führen, dass der Alarm zu empfindlich ist.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

5XXError

Abmessungen:ApiName, Stufe

Alarmbeschreibung: Dieser Alarm hilft dabei, eine hohe Rate serverseitiger Fehler zu erkennen. Dies kann darauf hindeuten, dass im API-Backend, im Netzwerk oder bei der Integration zwischen dem API-Gateway und der Backend-API etwas nicht stimmt. Diese Dokumentation kann Ihnen helfen, die Ursache von 5XX-Fehlern zu beheben.

Absicht: Dieser Alarm kann hohe Raten serverseitiger Fehler bei den API-Gateway-Anfragen erkennen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 0,05

Begründung des Schwellenwerts: Mit dem vorgeschlagenen Schwellenwert wird erkannt, wenn bei mehr als 5 % der gesamten Anfragen 5XX-Fehler auftreten. Sie können den Schwellenwert jedoch an den Datenverkehr der Anfragen sowie an die akzeptablen Fehlerraten anpassen. Sie können auch historische Daten analysieren, um die akzeptable Fehlerrate für den Anwendungs-Workload zu ermitteln, und den Schwellenwert dann entsprechend anpassen. Bei häufig auftretenden 5XX-Fehlern muss ein Alarm ausgelöst werden. Die Einstellung eines sehr niedrigen Schwellenwerts kann jedoch dazu führen, dass der Alarm zu empfindlich ist.

Zeitraum: 60

Datenpunkte bis Alarm: 3

Auswertungszeiträume: 3

Vergleichsoperator: GREATER_THAN_THRESHOLD

Count

Abmessungen: ApiName Bühne

Alarmbeschreibung: Dieser Alarm hilft dabei, ein geringes Datenverkehrsaufkommen für die REST-API-Phase zu erkennen. Dies kann ein Hinweis auf ein Problem mit der Anwendung sein, die die API aufruft, z. B. die Verwendung falscher Endpunkte. Es könnte auch ein Hinweis auf ein Problem mit der Konfiguration oder den Berechtigungen der API sein, so dass sie für Kunden nicht erreichbar ist.

Absicht: Dieser Alarm kann ein unerwartet geringes Verkehrsaufkommen in der REST-API-Phase erkennen. Wir empfehlen Ihnen, diesen Alarm zu erstellen, wenn Ihre API unter normalen Bedingungen eine vorhersehbare und konstante Anzahl von Anfragen erhält. Wenn Sie detaillierte CloudWatch Messwerte aktiviert haben und das normale Verkehrsaufkommen pro Methode und Ressource vorhersagen können, empfehlen wir Ihnen, alternative Alarme zu erstellen, um den Rückgang des Verkehrsaufkommens für jede Ressource und Methode genauer überwachen zu können. Dieser Alarm wird nicht für APIs empfohlen, die keinen konstanten und gleichmäßigen Datenverkehr erwarten.

Statistik: SampleCount

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Legen Sie den Schwellenwert auf der Grundlage historischer Datenanalysen fest, um zu ermitteln, wie viele Anfragen für Ihre API zu erwarten sind. Wenn Sie den Schwellenwert auf einen sehr hohen Wert setzen, kann dies dazu führen, dass der Alarm in Zeiten mit normalem und erwartungsgemäß geringem Datenverkehr zu empfindlich reagiert. Umgekehrt könnte eine Einstellung auf einen sehr niedrigen Wert dazu führen, dass der Alarm ungewöhnliche kleinere Rückgänge des Verkehrsaufkommens übersieht.

Zeitraum: 60

Datenpunkte bis Alarm: 10

Auswertungszeiträume: 10

Vergleichsoperator: LESS_THAN_THRESHOLD

Count

Dimensionen: PhaseApiName, Ressource, Methode

Alarmbeschreibung: Dieser Alarm hilft dabei, ein geringes Datenverkehrsaufkommen für die REST-API-Ressource und -Methode in der Phase zu erkennen. Dies kann auf ein Problem mit dem Aufruf der API durch die Anwendung hinweisen, z. B. die Verwendung falscher Endpunkte. Es könnte auch ein Hinweis auf ein Problem mit der Konfiguration oder den Berechtigungen der API sein, so dass sie für Kunden nicht erreichbar ist.

Absicht: Dieser Alarm kann ein unerwartet geringes Datenverkehrsaufkommen für die REST-API-Ressource und -Methode in der Phase erkennen. Wir empfehlen Ihnen, diesen Alarm zu erstellen, wenn Ihre API unter normalen Bedingungen eine vorhersehbare und konstante Anzahl von Anfragen erhält. Dieser Alarm wird nicht für APIs empfohlen, die keinen konstanten und gleichmäßigen Datenverkehr erwarten.

Statistik: SampleCount

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Legen Sie den Schwellenwert auf der Grundlage historischer Datenanalysen fest, um zu ermitteln, wie viele Anfragen für Ihre API zu erwarten sind. Wenn Sie den Schwellenwert auf einen sehr hohen Wert setzen, kann dies dazu führen, dass der Alarm in Zeiten mit normalem und erwartungsgemäß geringem Datenverkehr zu empfindlich reagiert. Umgekehrt könnte eine Einstellung auf einen sehr niedrigen Wert dazu führen, dass der Alarm ungewöhnliche kleinere Rückgänge des Verkehrsaufkommens übersieht.

Zeitraum: 60

Datenpunkte bis Alarm: 10

Auswertungszeiträume: 10

Vergleichsoperator: LESS_THAN_THRESHOLD

Count

Abmessungen:ApiId, Bühne

Alarmbeschreibung: Dieser Alarm hilft dabei, ein geringes Datenverkehrsaufkommen für die HTTP-API-Phase zu erkennen. Dies kann auf ein Problem mit dem Aufruf der API durch die Anwendung hinweisen, z. B. die Verwendung falscher Endpunkte. Es könnte auch ein Hinweis auf ein Problem mit der Konfiguration oder den Berechtigungen der API sein, so dass sie für Kunden nicht erreichbar ist.

Absicht: Dieser Alarm kann ein unerwartet geringes Datenverkehrsaufkommen in der HTTP-API-Phase erkennen. Wir empfehlen Ihnen, diesen Alarm zu erstellen, wenn Ihre API unter normalen Bedingungen eine vorhersehbare und konstante Anzahl von Anfragen erhält. Wenn Sie detaillierte CloudWatch Metriken aktiviert haben und das normale Verkehrsaufkommen pro Route vorhersagen können, empfehlen wir Ihnen, alternative Alarme zu erstellen, um den Rückgang des Verkehrsaufkommens für jede Route genauer überwachen zu können. Dieser Alarm wird nicht für APIs empfohlen, die keinen konstanten und gleichmäßigen Datenverkehr erwarten.

Statistik: SampleCount

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Legen Sie den Schwellenwert auf der Grundlage historischer Datenanalysen fest, um zu ermitteln, wie viele Anfragen für Ihre API zu erwarten sind. Wenn Sie den Schwellenwert auf einen sehr hohen Wert setzen, kann dies dazu führen, dass der Alarm in Zeiten mit normalem und erwartungsgemäß geringem Datenverkehr zu empfindlich reagiert. Umgekehrt könnte eine Einstellung auf einen sehr niedrigen Wert dazu führen, dass der Alarm ungewöhnliche kleinere Rückgänge des Verkehrsaufkommens übersieht.

Zeitraum: 60

Datenpunkte bis Alarm: 10

Auswertungszeiträume: 10

Vergleichsoperator: LESS_THAN_THRESHOLD

Count

Dimensionen: PhaseApiId, Ressource, Methode

Alarmbeschreibung: Dieser Alarm hilft dabei, ein geringes Datenverkehrsaufkommen für die HTTP-API-Route in dieser Phase zu erkennen. Dies kann auf ein Problem mit dem Aufruf der API durch die Anwendung hinweisen, z. B. die Verwendung falscher Endpunkte. Dies könnte auch auf ein Problem mit der Konfiguration oder den Berechtigungen der API hinweisen, sodass sie für Clients nicht erreichbar ist.

Absicht: Dieser Alarm kann ein unerwartet geringes Datenverkehrsaufkommen in der HTTP-API-Route in dieser Phase erkennen. Wir empfehlen Ihnen, diesen Alarm zu erstellen, wenn Ihre API unter normalen Bedingungen eine vorhersehbare und konstante Anzahl von Anfragen erhält. Dieser Alarm wird nicht für APIs empfohlen, die keinen konstanten und gleichmäßigen Datenverkehr erwarten.

Statistik: SampleCount

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Legen Sie den Schwellenwert auf der Grundlage historischer Datenanalysen fest, um zu ermitteln, wie viele Anfragen für Ihre API zu erwarten sind. Wenn Sie den Schwellenwert auf einen sehr hohen Wert setzen, kann dies dazu führen, dass der Alarm in Zeiten mit normalem und erwartungsgemäß geringem Datenverkehr zu empfindlich reagiert. Umgekehrt könnte eine Einstellung auf einen sehr niedrigen Wert dazu führen, dass der Alarm ungewöhnliche kleinere Rückgänge des Verkehrsaufkommens übersieht.

Zeitraum: 60

Datenpunkte bis Alarm: 10

Auswertungszeiträume: 10

Vergleichsoperator: LESS_THAN_THRESHOLD

IntegrationLatency

Abmessungen:ApiId, Bühne

Alarmbeschreibung: Dieser Alarm hilft zu erkennen, ob für die API-Anfragen in einer Phase eine hohe Integrationslatenz besteht. Sie können den `IntegrationLatency`-Metrikwert mit der entsprechenden Latenzmetrik Ihres Backends korrelieren, z. B. der `Duration`-Metrik für Lambda-Integrationen. Auf diese Weise können Sie feststellen, ob das API-Backend aufgrund von Leistungsproblemen mehr Zeit benötigt, um Anfragen von Clients zu verarbeiten, oder ob durch die Initialisierung oder den Kaltstart ein anderer Overhead entsteht. Erwägen Sie außerdem, CloudWatch Logs für Ihre API zu aktivieren und die Protokolle auf Fehler zu überprüfen, die zu den Problemen mit der hohen Latenz führen könnten. Erwägen Sie außerdem, detaillierte CloudWatch Metriken zu aktivieren, um einen Überblick über diese Metrik pro Route zu erhalten, damit Sie die Ursache der Integrationslatenz eingrenzen können.

Absicht: Dieser Alarm kann erkennen, wenn die API-Gateway-Anfragen in einer Phase eine hohe Integrationslatenz aufweisen. Wir empfehlen diesen Alarm für WebSocket APIs und halten ihn für HTTP-APIs für optional, da es dort bereits separate Alarmempfehlungen für die Latenzmetrik gibt. Wenn Sie detaillierte CloudWatch Metriken aktiviert haben und unterschiedliche Leistungsanforderungen für die Integrationslatenz pro Route haben, empfehlen wir Ihnen, alternative Alarme zu erstellen, um die Integrationslatenz für jede Route genauer überwachen zu können.

Statistik: p90

Empfohlener Schwellenwert: 2 000,0

Begründung des Schwellenwerts: Der vorgeschlagene Wert des Schwellenwerts funktioniert nicht für alle API-Workloads. Sie können ihn jedoch als Ausgangspunkt für den Schwellenwert verwenden. Anschließend können Sie je nach Workload und akzeptablen Latenz-, Leistungs- und SLA-Anforderungen für die API unterschiedliche Schwellenwerte auswählen. Wenn es für die API generell akzeptabel ist, eine höhere Latenz zu haben, legen Sie einen höheren Schwellenwert fest, um den Alarm weniger empfindlich zu machen. Wenn jedoch erwartet wird, dass die API Antworten nahezu in Echtzeit liefert, legen Sie einen niedrigeren Schwellenwert fest. Sie können auch historische Daten analysieren, um die erwartete Basislatenz für den Anwendungs-Workload zu ermitteln. Anhand dieser Daten können Sie dann den Schwellenwert entsprechend anpassen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

IntegrationLatency

Abmessungen:ApiId, Etappe, Route

Beschreibung des Alarms: Dieser Alarm hilft zu erkennen, ob bei WebSocket API-Anfragen für eine Route in einer Phase eine hohe Integrationslatenz besteht. Sie können den `IntegrationLatency`-Metrikwert mit der entsprechenden Latenzmetrik Ihres Backends korrelieren, z. B. der `Duration`-Metrik für Lambda-Integrationen. Auf diese Weise können Sie feststellen, ob das API-Backend aufgrund von Leistungsproblemen mehr Zeit benötigt, um Anfragen von Clients zu verarbeiten, oder ob durch die Initialisierung oder den Kaltstart ein anderer Overhead entsteht. Erwägen Sie außerdem, CloudWatch Logs für Ihre API zu aktivieren und die Protokolle auf Fehler zu überprüfen, die die Probleme mit der hohen Latenz verursachen könnten.

Absicht: Dieser Alarm kann erkennen, wenn die API-Gateway-Anfragen für eine Route in einer Phase eine hohe Integrationslatenz aufweisen.

Statistik: p90

Empfohlener Schwellenwert: 2 000,0

Begründung des Schwellenwerts: Der vorgeschlagene Wert des Schwellenwerts funktioniert nicht für alle API-Workloads. Sie können ihn jedoch als Ausgangspunkt für den Schwellenwert verwenden. Anschließend können Sie je nach Workload und akzeptablen Latenz-, Leistungs- und SLA-Anforderungen für die API unterschiedliche Schwellenwerte auswählen. Wenn es für die API generell akzeptabel ist, eine höhere Latenz zu haben, können Sie einen höheren Schwellenwert festlegen, um den Alarm weniger empfindlich zu machen. Wenn jedoch erwartet wird, dass die API Antworten nahezu in Echtzeit liefert, legen Sie einen niedrigeren Schwellenwert fest. Sie können auch historische Daten analysieren, um die erwartete Basislatenz für den Anwendungs-Workload zu ermitteln. Anhand dieser Daten können Sie dann den Schwellenwert entsprechend anpassen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latency

Abmessungen:ApiName, Bühne

Alarmbeschreibung: Dieser Alarm erkennt eine hohe Latenz in einer Phase. Suchen Sie den `IntegrationLatency`-Metrikwert, um die Latenz des API-Backends zu überprüfen. Wenn die beiden Metriken größtenteils übereinstimmen, ist das API-Backend die Ursache für die höhere Latenz, und Sie sollten dort nach Problemen suchen. Erwägen Sie auch, CloudWatch Protokolle zu aktivieren und nach Fehlern zu suchen, die die hohe Latenz verursachen könnten. Erwägen Sie außerdem, detaillierte CloudWatch Metriken zu aktivieren, um diese Metrik pro Ressource und Methode anzuzeigen und die Ursache der Latenz einzugrenzen. Falls zutreffend, lesen Sie die Anleitungen zur Fehlerbehebung mit Lambda oder zur Fehlerbehebung für Edge-optimierte API-Endpunkte.

Absicht: Dieser Alarm kann erkennen, wenn die API-Gateway-Anfragen in einer Phase eine hohe Latenz aufweisen. Wenn Sie detaillierte CloudWatch Metriken aktiviert haben und für jede Methode und Ressource unterschiedliche Anforderungen an die Latenzleistung haben, empfehlen wir Ihnen, alternative Alarme zu erstellen, um die Latenz für jede Ressource und Methode genauer überwachen zu können.

Statistik: p90

Empfohlener Schwellenwert: 2 500,0

Begründung des Schwellenwerts: Der vorgeschlagene Schwellenwert funktioniert nicht für alle API-Workloads. Sie können ihn jedoch als Ausgangspunkt für den Schwellenwert verwenden. Anschließend können Sie je nach Workload und akzeptablen Latenz-, Leistungs- und SLA-Anforderungen für die API unterschiedliche Schwellenwerte auswählen. Wenn es für die API generell akzeptabel ist, eine höhere Latenz zu haben, können Sie einen höheren Schwellenwert festlegen, um den Alarm weniger empfindlich zu machen. Wenn jedoch erwartet wird, dass die API Antworten nahezu in Echtzeit liefert, legen Sie einen niedrigeren Schwellenwert fest. Sie können auch historische Daten analysieren, um die erwartete Basislatenz für den Anwendungs-Workload zu ermitteln, und dann den Schwellenwert entsprechend anpassen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latency

Dimensionen: PhaseApiName, Ressource, Methode

Alarmbeschreibung: Dieser Alarm erkennt eine hohe Latenz für eine Ressource und Methode in einer Phase. Suchen Sie den `IntegrationLatency`-Metrikwert, um die Latenz des API-Backends zu überprüfen. Wenn die beiden Metriken größtenteils übereinstimmen, ist das API-Backend die Ursache für die höhere Latenz, und Sie sollten dort nach Leistungsproblemen suchen. Erwägen Sie auch, CloudWatch Protokolle zu aktivieren und nach Fehlern zu suchen, die die hohe Latenz verursachen könnten. Falls zutreffend, finden Sie auch die Anleitungen zur Fehlerbehebung mit Lambda oder zur Fehlerbehebung für Edge-optimierte API-Endpunkte.

Absicht: Dieser Alarm kann erkennen, wenn die API-Gateway-Anfragen für eine Ressource und Methode in einer Phase eine hohe Latenz aufweisen.

Statistik: p90

Empfohlener Schwellenwert: 2 500,0

Begründung des Schwellenwerts: Der vorgeschlagene Wert des Schwellenwerts funktioniert nicht für alle API-Workloads. Sie können ihn jedoch als Ausgangspunkt für den Schwellenwert verwenden. Anschließend können Sie je nach Workload und akzeptablen Latenz-, Leistungs- und

SLA-Anforderungen für die API unterschiedliche Schwellenwerte auswählen. Wenn es für die API generell akzeptabel ist, eine höhere Latenz zu haben, können Sie einen höheren Schwellenwert festlegen, um den Alarm weniger empfindlich zu machen. Wenn jedoch erwartet wird, dass die API Antworten nahezu in Echtzeit liefert, legen Sie einen niedrigeren Schwellenwert fest. Sie können auch historische Daten analysieren, um die erwartete Basislatenz für den Anwendungs-Workload zu ermitteln, und dann den Schwellenwert entsprechend anpassen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latency

Abmessungen:ApiId, Bühne

Alarmbeschreibung: Dieser Alarm erkennt eine hohe Latenz in einer Phase. Suchen Sie den `IntegrationLatency`-Metrikwert, um die Latenz des API-Backends zu überprüfen. Wenn die beiden Metriken größtenteils übereinstimmen, ist das API-Backend die Ursache für die höhere Latenz, und Sie sollten dort nach Leistungsproblemen suchen. Erwägen Sie auch, CloudWatch Logs zu aktivieren und nach Fehlern zu suchen, die die hohe Latenz verursachen könnten. Erwägen Sie außerdem, detaillierte CloudWatch Metriken zu aktivieren, um diese Metrik pro Route anzuzeigen und die Ursache der Latenz einzugrenzen. Falls zutreffend, können Sie auch den Leitfaden zur Fehlerbehebung mit Lambda-Integrationen lesen.

Absicht: Dieser Alarm kann erkennen, wenn die API-Gateway-Anfragen in einer Phase eine hohe Latenz aufweisen. Wenn Sie detaillierte CloudWatch Metriken aktiviert haben und unterschiedliche Anforderungen an die Latenzleistung pro Route haben, empfehlen wir Ihnen, alternative Alarme zu erstellen, um die Latenz für jede Route genauer überwachen zu können.

Statistik: p90

Empfohlener Schwellenwert: 2 500,0

Begründung des Schwellenwerts: Der vorgeschlagene Wert des Schwellenwerts funktioniert nicht für alle API-Workloads. Er kann jedoch als Ausgangspunkt für den Schwellenwert verwendet werden. Anschließend können Sie je nach Workload und akzeptablen Latenz-, Leistungs- und SLA-Anforderungen für die API unterschiedliche Schwellenwerte auswählen. Wenn es akzeptabel ist, dass die API generell eine höhere Latenz aufweist, können Sie einen höheren Schwellenwert

festlegen, um sie weniger empfindlich zu machen. Wenn jedoch erwartet wird, dass die API Antworten nahezu in Echtzeit liefert, legen Sie einen niedrigeren Schwellenwert fest. Sie können auch historische Daten analysieren, um die erwartete Basislatenz für den Anwendungs-Workload zu ermitteln, und dann den Schwellenwert entsprechend anpassen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latency

Dimensionen: PhaseApiId, Ressource, Methode

Alarmbeschreibung: Dieser Alarm erkennt eine hohe Latenz für eine Route in einer Phase. Suchen Sie den `IntegrationLatency`-Metrikwert, um die Latenz des API-Backends zu überprüfen. Wenn die beiden Metriken größtenteils übereinstimmen, ist das API-Backend die Ursache für die höhere Latenz und sollte auf Leistungsprobleme untersucht werden. Erwägen Sie auch, CloudWatch Protokolle zu aktivieren und nach Fehlern zu suchen, die die hohe Latenz verursachen könnten. Falls zutreffend, können Sie auch den Leitfaden zur Fehlerbehebung mit Lambda-Integrationen lesen.

Absicht: Dieser Alarm wird verwendet, um zu erkennen, wenn die API-Gateway-Anfragen für eine Route in einer Phase eine hohe Latenz aufweisen.

Statistik: p90

Empfohlener Schwellenwert: 2 500,0

Begründung des Schwellenwerts: Der vorgeschlagene Wert des Schwellenwerts funktioniert nicht für alle API-Workloads. Er kann jedoch als Ausgangspunkt für den Schwellenwert verwendet werden. Anschließend können Sie je nach Workload und akzeptablen Latenz-, Leistungs- und SLA-Anforderungen für die API unterschiedliche Schwellenwerte auswählen. Wenn es für die API generell akzeptabel ist, eine höhere Latenz zu haben, können Sie einen höheren Schwellenwert festlegen, um den Alarm weniger empfindlich zu machen. Wenn jedoch erwartet wird, dass die API Antworten nahezu in Echtzeit liefert, legen Sie einen niedrigeren Schwellenwert fest. Sie können auch historische Daten analysieren, um die erwartete Basislatenz für den Anwendungs-Workload zu ermitteln, und dann den Schwellenwert entsprechend anpassen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

4xx

Abmessungen:ApiId, Bühne

Alarmbeschreibung: Dieser Alarm erkennt eine hohe Rate von clientseitigen Fehlern. Dies kann auf ein Problem mit den Autorisierungs- oder Client-Anfrageparametern hinweisen. Es könnte auch bedeuten, dass eine Route entfernt wurde oder ein Client eine anfordert, die in der API nicht vorhanden ist. Erwägen Sie, CloudWatch Logs zu aktivieren und nach Fehlern zu suchen, die die 4xx-Fehler verursachen könnten. Erwägen Sie außerdem, detaillierte CloudWatch Metriken zu aktivieren, um diese Metrik pro Route anzuzeigen, damit Sie die Fehlerquelle eingrenzen können. Fehler können auch durch eine Überschreitung des konfigurierten Drosselungslimits verursacht werden. Wenn in den Antworten und Protokollen eine hohe und unerwartete Anzahl von 429-Fehlern gemeldet wird, folgen Sie dieser Anleitung, um dieses Problem zu beheben.

Absicht: Dieser Alarm kann eine hohe Anzahl von clientseitigen Fehlern bei den API-Gateway-Anfragen erkennen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 0,05

Begründung des Schwellenwerts: Mit dem vorgeschlagenen Schwellenwert wird erkannt, wenn bei mehr als 5 % der gesamten Anfragen 4XX-Fehler auftreten. Sie können den Schwellenwert jedoch so einstellen, dass er dem Datenverkehr der Anfragen sowie den akzeptablen Fehlerraten entspricht. Sie können auch historische Daten analysieren, um die akzeptable Fehlerrate für den Anwendungs-Workload zu bestimmen, und den Schwellenwert dann entsprechend anpassen. Bei häufig auftretenden 4XX-Fehlern muss ein Alarm ausgelöst werden. Die Einstellung eines sehr niedrigen Schwellenwerts kann jedoch dazu führen, dass der Alarm zu empfindlich ist.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

5xx

Abmessungen:ApiId, Stufe

Alarmbeschreibung: Dieser Alarm hilft dabei, eine hohe Rate serverseitiger Fehler zu erkennen. Dies kann darauf hindeuten, dass im API-Backend, im Netzwerk oder bei der Integration zwischen dem API-Gateway und der Backend-API etwas nicht stimmt. Diese Dokumentation kann Ihnen helfen, die Ursache für 5XX-Fehler zu beheben.

Absicht: Dieser Alarm kann hohe Raten serverseitiger Fehler bei den API-Gateway-Anfragen erkennen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 0,05

Begründung des Schwellenwerts: Mit dem vorgeschlagenen Schwellenwert wird erkannt, wenn bei mehr als 5 % der gesamten Anfragen 5XX-Fehler auftreten. Sie können den Schwellenwert jedoch so einstellen, dass er dem Datenverkehr der Anfragen sowie den akzeptablen Fehlerraten entspricht. Sie können auch historische Daten analysieren, um die akzeptable Fehlerrate für den Anwendungs-Workload zu ermitteln. Anschließend können Sie den Schwellenwert entsprechend anpassen. Bei häufig auftretenden 5XX-Fehlern muss ein Alarm ausgelöst werden. Die Einstellung eines sehr niedrigen Schwellenwerts kann jedoch dazu führen, dass der Alarm zu empfindlich ist.

Zeitraum: 60

Datenpunkte bis Alarm: 3

Auswertungszeiträume: 3

Vergleichsoperator: GREATER_THAN_THRESHOLD

MessageCount

Abmessungen: ApiId Bühne

Beschreibung des Alarms: Dieser Alarm hilft dabei, ein geringes Verkehrsaufkommen in der WebSocket API-Phase zu erkennen. Dies kann auf ein Problem beim Aufruf der API durch Clients hinweisen, z. B. die Verwendung falscher Endpunkte oder Probleme mit dem Backend, das

Nachrichten an Clients sendet. Dies könnte auch auf ein Problem mit der Konfiguration oder den Berechtigungen der API hinweisen, sodass sie für Clients nicht erreichbar ist.

Absicht: Dieser Alarm kann ein unerwartet geringes Verkehrsaufkommen in der WebSocket API-Phase erkennen. Wir empfehlen Ihnen, diesen Alarm zu erstellen, wenn Ihre API unter normalen Bedingungen eine vorhersehbare und konsistente Anzahl von Nachrichten empfängt und sendet. Wenn Sie detaillierte CloudWatch Metriken aktiviert haben und das normale Verkehrsaufkommen pro Route vorhersagen können, ist es besser, alternative Alarme zu diesem zu erstellen, um den Rückgang des Verkehrsaufkommens für jede Route genauer überwachen zu können. Wir empfehlen diesen Alarm nicht für APIs, die keinen konstanten und konsistenten Datenverkehr erwarten.

Statistik: SampleCount

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Legen Sie den Schwellenwert auf der Grundlage historischer Datenanalysen fest, um zu ermitteln, wie hoch die erwartete Anzahl an Nachrichten für Ihre API ist. Wenn Sie den Schwellenwert auf einen sehr hohen Wert setzen, kann dies dazu führen, dass der Alarm in Zeiten mit normalem und erwartungsgemäß geringem Datenverkehr zu empfindlich reagiert. Umgekehrt könnte eine Einstellung auf einen sehr niedrigen Wert dazu führen, dass der Alarm ungewöhnliche kleinere Rückgänge des Verkehrsaufkommens übersieht.

Zeitraum: 60

Datenpunkte bis Alarm: 10

Auswertungszeiträume: 10

Vergleichsoperator: LESS_THAN_THRESHOLD

MessageCount

Abmessungen:ApiId, Etappe, Route

Beschreibung des Alarms: Dieser Alarm hilft dabei, ein geringes Verkehrsaufkommen für die WebSocket API-Route in der Phase zu erkennen. Dies kann auf ein Problem beim Aufrufen der API durch die Clients hinweisen, z. B. auf die Verwendung falscher Endpunkte oder auf Probleme beim Senden von Nachrichten an Clients durch das Backend. Dies könnte auch auf ein Problem mit der Konfiguration oder den Berechtigungen der API hinweisen, sodass sie für Clients nicht erreichbar ist.

Absicht: Dieser Alarm kann ein unerwartet geringes Verkehrsaufkommen für die WebSocket API-Route in der Phase erkennen. Wir empfehlen Ihnen, diesen Alarm zu erstellen, wenn Ihre API unter normalen Bedingungen eine vorhersehbare und konsistente Anzahl von Nachrichten empfängt und sendet. Wir empfehlen diesen Alarm nicht für APIs, die keinen konstanten und konsistenten Datenverkehr erwarten.

Statistik: SampleCount

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Legen Sie den Schwellenwert auf der Grundlage historischer Datenanalysen fest, um zu ermitteln, wie hoch die erwartete Anzahl an Nachrichten für Ihre API ist. Wenn Sie den Schwellenwert auf einen sehr hohen Wert setzen, kann dies dazu führen, dass der Alarm in Zeiten mit normalem und erwartungsgemäß geringem Datenverkehr zu empfindlich reagiert. Umgekehrt könnte eine Einstellung auf einen sehr niedrigen Wert dazu führen, dass der Alarm ungewöhnliche kleinere Rückgänge des Verkehrsaufkommens übersieht.

Zeitraum: 60

Datenpunkte bis Alarm: 10

Auswertungszeiträume: 10

Vergleichsoperator: LESS_THAN_THRESHOLD

ClientError

Abmessungen:ApiId, Bühne

Alarmbeschreibung: Dieser Alarm erkennt eine hohe Rate von Client-Fehlern. Dies kann auf ein Problem mit den Autorisierungs- oder Nachrichtenparametern hinweisen. Es könnte auch bedeuten, dass eine Route entfernt wurde oder ein Client eine anfordert, die in der API nicht vorhanden ist. Erwägen Sie, CloudWatch Logs zu aktivieren und nach Fehlern zu suchen, die die 4xx-Fehler verursachen könnten. Erwägen Sie außerdem, detaillierte CloudWatch Metriken zu aktivieren, um diese Metrik pro Route anzuzeigen, damit Sie die Fehlerquelle eingrenzen können. Fehler können auch durch eine Überschreitung des konfigurierten Drosselungslimits verursacht werden. Wenn in den Antworten und Protokollen eine hohe und unerwartete Anzahl von 429-Fehlern gemeldet wird, folgen Sie dieser Anleitung, um dieses Problem zu beheben.

Absicht: Dieser Alarm kann eine hohe Anzahl von Client-Fehlern für die WebSocket API-Gateway-Nachrichten erkennen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 0,05

Begründung des Schwellenwerts: Mit dem vorgeschlagenen Schwellenwert wird erkannt, wenn bei mehr als 5 % der gesamten Anfragen 4XX-Fehler auftreten. Sie können den Schwellenwert sowohl an den Datenverkehr der Anfragen als auch an Ihre akzeptablen Fehlerraten anpassen. Sie können auch historische Daten analysieren, um die akzeptable Fehlerrate für den Anwendungs-Workload zu ermitteln, und dann den Schwellenwert entsprechend anpassen. Bei häufig auftretenden 4XX-Fehlern muss ein Alarm ausgelöst werden. Die Einstellung eines sehr niedrigen Schwellenwerts kann jedoch dazu führen, dass der Alarm zu empfindlich ist.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

ExecutionError

Abmessungen:ApiId, Bühne

Alarmbeschreibung: Dieser Alarm hilft dabei, eine hohe Anzahl von Ausführungsfehlern zu erkennen. Dies kann durch 5XX-Fehler aufgrund Ihrer Integration, durch Berechtigungsprobleme oder durch andere Faktoren verursacht werden, die einen erfolgreichen Aufruf der Integration verhindern, z. B. wenn die Integration gedrosselt oder gelöscht wird. Erwägen Sie, CloudWatch Logs für Ihre API zu aktivieren und die Logs auf Art und Ursache der Fehler zu überprüfen. Erwägen Sie außerdem, detaillierte CloudWatch Metriken zu aktivieren, um einen Überblick über diese Metrik pro Route zu erhalten und so die Fehlerquelle einzugrenzen. Diese Dokumentation kann Ihnen auch dabei helfen, die Ursache von Verbindungsfehlern zu beheben.

Absicht: Dieser Alarm kann eine hohe Anzahl von Ausführungsfehlern für die WebSocket API-Gateway-Nachrichten erkennen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 0,05

Begründung des Schwellenwerts: Der vorgeschlagene Schwellenwert erkennt, wenn bei mehr als 5 % der gesamten Anfragen Ausführungsfehler auftreten. Sie können den Schwellenwert

sowohl an den Datenverkehr der Anfragen als auch an Ihre akzeptablen Fehlerraten anpassen. Sie können historische Daten analysieren, um die akzeptable Fehlerrate für den Anwendungs-Workload zu ermitteln, und dann den Schwellenwert entsprechend anpassen. Bei häufig auftretenden Ausführungsfehlern muss ein Alarm ausgelöst werden. Die Einstellung eines sehr niedrigen Schwellenwerts kann jedoch dazu führen, dass der Alarm zu empfindlich ist.

Zeitraum: 60

Datenpunkte bis Alarm: 3

Auswertungszeiträume: 3

Vergleichsoperator: GREATER_THAN_THRESHOLD

Amazon EC2 Auto Scaling

GroupInServiceCapacity

Abmessungen: AutoScalingGroupName

Alarmbeschreibung: Dieser Alarm hilft zu erkennen, wenn die Kapazität in der Gruppe unter der gewünschten Kapazität liegt, die für Ihren Workload erforderlich ist. Um Fehler zu beheben, überprüfen Sie Ihre Skalierungsaktivitäten auf Startfehler und stellen Sie sicher, dass Ihre gewünschte Kapazitätskonfiguration korrekt ist.

Absicht: Dieser Alarm kann eine geringe Verfügbarkeit in Ihrer Auto-Scaling-Gruppe aufgrund von Startfehlern oder unterbrochenen Starts erkennen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Der Schwellenwert sollte der Mindestkapazität entsprechen, die zur Ausführung Ihres Workloads erforderlich ist. In den meisten Fällen können Sie dies so einstellen, dass es der GroupDesiredCapacity Metrik entspricht.

Zeitraum: 60

Datenpunkte bis Alarm: 10

Auswertungszeiträume: 10

Vergleichsoperator: LESS_THAN_THRESHOLD

Amazon CloudFront

5 xxErrorRate

Abmessungen:DistributionId, Region=Global

Beschreibung des Alarms: Dieser Alarm überwacht den Prozentsatz der 5xx-Fehlerantworten von Ihrem Ursprungsserver, damit Sie feststellen können, ob der CloudFront Dienst Probleme hat. Unter Fehlerbehebung bei Fehlerantworten von Ihrem Ursprungsserver finden Sie Informationen, die Ihnen helfen, die Probleme mit Ihrem Server zu verstehen. Außerdem können Sie zusätzliche Metriken aktivieren, um detaillierte Fehlermetriken zu erhalten.

Absicht: Dieser Alarm wird verwendet, um Probleme mit der Bearbeitung von Anfragen vom Ursprungsserver oder Probleme mit der Kommunikation zwischen CloudFront und Ihrem Ursprungsserver zu erkennen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Der empfohlene Schwellenwert für diesen Alarm hängt stark von der Toleranz für 5XX-Antworten ab. Sie können historische Daten und Trends analysieren und dann den Schwellenwert entsprechend festlegen. Da 5XX-Fehler durch vorübergehende Probleme verursacht werden können, empfehlen wir, den Schwellenwert auf einen Wert größer als 0 festzulegen, damit der Alarm nicht zu empfindlich reagiert.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

OriginLatency

Abmessungen:DistributionId, Region=Global

Alarmbeschreibung: Der Alarm hilft zu überwachen, ob der Ursprungsserver zu lange braucht, um zu antworten. Wenn der Server zu lange braucht, um zu antworten, kann dies zu einem Timeout

führen. Lesen Sie, wie Sie verzögerte Antworten von Anwendungen auf Ihrem Ursprungsserver finden und beheben können, wenn Sie konstant hohe `OriginLatency`-Werte feststellen.

Absicht: Dieser Alarm wird verwendet, um Probleme zu erkennen, bei denen der Ursprungsserver zu lange braucht, um zu antworten.

Statistik: p90

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Sie sollten den Wert von etwa 80 % des ursprünglichen Antwort-Timeouts berechnen und das Ergebnis als Schwellenwert verwenden. Wenn diese Metrik durchweg in der Nähe des Timeout-Werts für die Ursprungsserver-Antwort liegt, treten möglicherweise 504-Fehler auf.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

FunctionValidationErrors

Dimensionen:DistributionId,, Region=Global FunctionName

Beschreibung des Alarms: Dieser Alarm hilft Ihnen dabei, Validierungsfehler von CloudFront Funktionen zu überwachen, sodass Sie Maßnahmen ergreifen können, um sie zu beheben. Analysieren Sie die CloudWatch Funktionsprotokolle und schauen Sie sich den Funktionscode an, um die Ursache des Problems zu finden und zu beheben. Informationen zu den häufigsten Fehlkonfigurationen von Funktionen finden Sie unter Einschränkungen für CloudFront Edge-Funktionen.

Absicht: Dieser Alarm wird verwendet, um Validierungsfehler von CloudFront Funktionen zu erkennen.

Statistik: Summe

Empfohlener Schwellenwert: 0,0

Begründung des Schwellenwerts: Ein Wert über 0 weist auf einen Validierungsfehler hin. Wir empfehlen, den Schwellenwert auf 0 zu setzen, da Validierungsfehler auf ein Problem hinweisen,

wenn CloudFront Funktionen wieder an übergeben CloudFront werden. Benötigt beispielsweise CloudFront den HTTP-Host-Header, um eine Anfrage zu verarbeiten. Nichts hindert einen Benutzer daran, den Host-Header in seinem CloudFront Funktionscode zu löschen. Aber wenn die Antwort CloudFront zurückkommt und der Host-Header fehlt, CloudFront wird ein Validierungsfehler ausgegeben.

Zeitraum: 60

Datenpunkte bis Alarm: 2

Auswertungszeiträume: 2

Vergleichsoperator: GREATER_THAN_THRESHOLD

FunctionExecutionErrors

Abmessungen:DistributionId, FunctionName, Region=Global

Beschreibung des Alarms: Dieser Alarm hilft Ihnen dabei, Ausführungsfehler von CloudFront Funktionen zu überwachen, sodass Sie Maßnahmen ergreifen können, um sie zu beheben. Analysieren Sie die CloudWatch Funktionsprotokolle und schauen Sie sich den Funktionscode an, um die Ursache des Problems zu finden und zu beheben.

Absicht: Dieser Alarm wird verwendet, um Ausführungsfehler von CloudFront Funktionen zu erkennen.

Statistik: Summe

Empfohlener Schwellenwert: 0,0

Begründung des Schwellenwerts: Wir empfehlen, den Schwellenwert auf 0 zu setzen, da ein Ausführungsfehler auf ein Problem mit dem Code hinweist, das zur Laufzeit auftritt.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

FunctionThrottles

Dimensionen:DistributionId, FunctionName, Region=Global

Beschreibung des Alarms: Dieser Alarm hilft Ihnen zu überwachen, ob Ihre CloudFront Funktion gedrosselt ist. Wenn Ihre Funktion gedrosselt ist, bedeutet dies, dass die Ausführung zu lange dauert. Um Funktionsdrosselungen zu vermeiden, sollten Sie eine Optimierung des Funktionscodes in Betracht ziehen.

Absicht: Dieser Alarm kann erkennen, wenn Ihre CloudFront Funktion gedrosselt ist, sodass Sie reagieren und das Problem lösen können, um ein reibungsloses Kundenerlebnis zu gewährleisten.

Statistik: Summe

Empfohlener Schwellenwert: 0,0

Begründung des Schwellenwerts: Wir empfehlen, den Schwellenwert auf 0 zu setzen, um eine schnellere Auflösung der Funktionsdrosselung zu ermöglichen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

Amazon Cognito

SignUpThrottles

Abmessungen:, UserPool UserPoolClient

Alarmbeschreibung: Dieser Alarm überwacht die Anzahl der gedrosselten Anfragen. Wenn Benutzer ständig gedrosselt werden, sollten Sie das Limit erhöhen, indem Sie eine Erhöhung der Service Quotas beantragen. In Kontingente in Amazon Cognito erfahren Sie, wie Sie eine Kontingenterhöhung beantragen können. Um proaktiv Maßnahmen zu ergreifen, sollten Sie die Nutzungsquote nachverfolgen.

Absicht: Dieser Alarm hilft dabei, das Auftreten gedrosselter Registrierungsanfragen zu überwachen. Auf diese Weise können Sie wissen, wann Sie Maßnahmen ergreifen müssen, um eine Verschlechterung des Anmeldeerlebnisses zu verhindern. Anhaltende Drosselung von Anfragen wirkt sich negativ auf die Anmeldeerfahrung der Benutzer aus.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Bei einem gut ausgestatteten Benutzerpool sollte es zu keiner Drosselung kommen, die sich über mehrere Datenpunkte erstreckt. Ein typischer Schwellenwert für einen erwarteten Workload sollte also Null sein. Bei einem unregelmäßigen Workload mit häufigen Spitzenwerten können Sie historische Daten analysieren, um die akzeptable Drosselung für den Anwendungs-Workload zu ermitteln, und dann den Schwellenwert entsprechend anpassen. Eine gedrosselte Anfrage sollte erneut versucht werden, um die Auswirkungen auf die Anwendung so gering wie möglich zu halten.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

SignInThrottles

Abmessungen:UserPool, UserPoolClient

Alarmbeschreibung: Dieser Alarm überwacht die Anzahl der gedrosselten Benutzerauthentifizierungsanfragen. Wenn Benutzer ständig gedrosselt werden, müssen Sie das Limit möglicherweise erhöhen, indem Sie eine Erhöhung der Service Quotas beantragen. In Kontingente in Amazon Cognito erfahren Sie, wie Sie eine Kontingenterhöhung beantragen können. Um proaktiv Maßnahmen zu ergreifen, sollten Sie die Nutzungsquote nachverfolgen.

Absicht: Dieser Alarm hilft dabei, das Auftreten gedrosselter Anmeldeanfragen zu überwachen. Auf diese Weise können Sie wissen, wann Sie Maßnahmen ergreifen müssen, um eine Verschlechterung der Anmeldeerfahrung zu verhindern. Eine anhaltende Drosselung von Anfragen führt zu einer schlechten Benutzererfahrung bei der Authentifizierung.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Bei einem gut ausgestatteten Benutzerpool sollte es zu keiner Drosselung kommen, die sich über mehrere Datenpunkte erstreckt. Ein typischer Schwellenwert für einen erwarteten Workload sollte also Null sein. Bei einem unregelmäßigen Workload mit

häufigen Spitzenwerten können Sie historische Daten analysieren, um die akzeptable Drosselung für den Anwendungs-Workload zu ermitteln, und dann den Schwellenwert entsprechend anpassen. Eine gedrosselte Anfrage sollte erneut versucht werden, um die Auswirkungen auf die Anwendung so gering wie möglich zu halten.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

TokenRefreshThrottles

Abmessungen:UserPool, UserPoolClient

Alarmbeschreibung: Sie können den Schwellenwert so einstellen, dass er sowohl dem Datenverkehr der Anfrage als auch einer akzeptablen Drosselung für Token-Aktualisierungsanfragen entspricht. Die Drosselung wird verwendet, um Ihr System vor zu vielen Anfragen zu schützen. Es ist jedoch wichtig zu überwachen, ob Sie auch für Ihren normalen Datenverkehr nicht ausreichend ausgestattet sind. Sie können historische Daten analysieren, um die akzeptable Drosselung für den Anwendungs-Workloads zu ermitteln, und dann den Alarmschwellenwert so einstellen, dass er über dem akzeptablen Drosselungswert liegt. Gedrosselte Anfragen sollten von der Anwendung/dem Service erneut versucht werden, da sie kurzlebig sind. Daher kann ein sehr niedriger Wert für den Schwellenwert dazu führen, dass der Alarm empfindlich ist.

Absicht: Dieser Alarm hilft dabei, das Auftreten gedrosselter Token-Aktualisierungsanfragen zu überwachen. Auf diese Weise wissen Sie, wann Sie Maßnahmen ergreifen müssen, um potenzielle Probleme zu beheben, eine reibungslose Benutzererfahrung sowie die Integrität und Zuverlässigkeit Ihres Authentifizierungssystems zu gewährleisten. Eine anhaltende Drosselung von Anfragen führt zu einer schlechten Benutzererfahrung bei der Authentifizierung.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Der Schwellenwert kann auch so eingestellt/angepasst werden, dass er dem Datenverkehr der Anfrage entspricht, sowie eine akzeptable Drosselung für Token-Aktualisierungsanfragen bietet. Drosselungen dienen dazu, Ihr System vor zu vielen Anfragen

zu schützen. Es ist jedoch wichtig, zu überwachen, ob Sie auch für Ihren normalen Datenverkehr nicht ausreichend ausgestattet sind, und zu prüfen, ob dies die Auswirkungen verursacht. Historische Daten können auch analysiert werden, um festzustellen, welche Drosselung für den Anwendungs-Workload akzeptabel ist. Der Schwellenwert kann auch höher eingestellt werden als die übliche zulässige Drosselungsstufe. Gedrosselte Anfragen sollten von der Anwendung/dem Service erneut versucht werden, da sie kurzlebig sind. Daher kann ein sehr niedriger Wert für den Schwellenwert dazu führen, dass der Alarm empfindlich ist.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

FederationThrottles

Abmessungen:UserPool, UserPoolClient, IdentityProvider

Alarmbeschreibung: Dieser Alarm überwacht die Anzahl der gedrosselten Identity Identitätsverbund-Anfragen. Wenn Sie ständig eine Drosselung feststellen, deutet dies möglicherweise darauf hin, dass Sie das Limit erhöhen müssen, indem Sie eine Erhöhung der Service Quotas beantragen. In Kontingente in Amazon Cognito erfahren Sie, wie Sie eine Kontingenterhöhung beantragen können.

Absicht: Dieser Alarm hilft dabei, das Auftreten gedrosselter Identitätsverbund-Anfragen zu überwachen. Dies kann Ihnen helfen, proaktiv auf Leistungsengpässe oder Fehlkonfigurationen zu reagieren und eine reibungslose Authentifizierung für Ihre Benutzer sicherzustellen. Eine anhaltende Drosselung von Anfragen führt zu einer schlechten Benutzererfahrung bei der Authentifizierung.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Sie können den Schwellenwert so festlegen, dass er sowohl dem Datenverkehr der Anfrage als auch der akzeptablen Drosselung für Identitätsverbund-Anfragen entspricht. Die Drosselung wird verwendet, um Ihr System vor zu vielen Anfragen zu schützen. Es ist jedoch wichtig zu überwachen, ob Sie auch für Ihren normalen Datenverkehr nicht ausreichend ausgestattet sind. Sie können historische Daten analysieren, um die akzeptable

Drosselung für den Anwendungs-Workload zu ermitteln, und dann den Schwellenwert auf einen Wert festlegen, der über Ihrer akzeptablen Drosselungsstufe liegt. Gedrosselte Anfragen sollten von der Anwendung/dem Service erneut versucht werden, da sie kurzlebig sind. Daher kann ein sehr niedriger Wert für den Schwellenwert dazu führen, dass der Alarm empfindlich ist.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

Amazon DynamoDB

AccountProvisionedReadCapacityUtilization

Dimensionen: Keine

Alarmbeschreibung: Dieser Alarm erkennt, ob die Lesekapazität des Kontos das bereitgestellte Limit erreicht. In diesem Fall können Sie das Kontingent für die Nutzung der Lesekapazität erhöhen. Mithilfe von Service Quotas können Sie Ihre aktuellen Kontingente für Lesekapazitätseinheiten einsehen und Erhöhungen beantragen.

Absicht: Der Alarm kann erkennen, ob sich die Lesekapazitätsauslastung des Kontos der bereitgestellten Lesekapazitätsauslastung nähert. Wenn die Auslastung ihr maximales Limit erreicht, beginnt DynamoDB, Leseanfragen zu drosseln.

Statistik: Maximum

Empfohlener Schwellenwert: 80,0

Begründung des Schwellenwerts: Legen Sie den Schwellenwert auf 80 % fest, sodass Maßnahmen (z. B. die Erhöhung der Kontolimits) ergriffen werden können, bevor die volle Kapazität erreicht ist, um eine Drosselung zu vermeiden.

Zeitraum: 300

Datenpunkte bis Alarm: 2

Auswertungszeiträume: 2

Vergleichsoperator: GREATER_THAN_THRESHOLD

AccountProvisionedWriteCapacityUtilization

Dimensionen: Keine

Alarmbeschreibung: Dieser Alarm erkennt, ob die Schreibkapazität des Kontos das bereitgestellte Limit erreicht. In diesem Fall können Sie das Kontingent für die Schreibkapazitätsnutzung erhöhen. Mithilfe von Service Quotas können Sie Ihre aktuellen Kontingente für Schreibkapazitätseinheiten einsehen und Erhöhungen beantragen.

Absicht: Dieser Alarm kann erkennen, ob sich die Schreibkapazitätsauslastung des Kontos der bereitgestellten Schreibkapazitätsauslastung nähert. Wenn die Auslastung ihr maximales Limit erreicht, beginnt DynamoDB, Schreibanfragen zu drosseln.

Statistik: Maximum

Empfohlener Schwellenwert: 80,0

Begründung des Schwellenwerts: Legen Sie den Schwellenwert auf 80 % fest, sodass Maßnahmen (z. B. die Erhöhung der Kontolimits) ergriffen werden können, bevor die volle Kapazität erreicht ist, um eine Drosselung zu vermeiden.

Zeitraum: 300

Datenpunkte bis Alarm: 2

Auswertungszeiträume: 2

Vergleichsoperator: GREATER_THAN_THRESHOLD

AgeOfOldestUnreplicatedRecord

Abmessungen:TableName, DelegatedOperation

Alarmbeschreibung: Dieser Alarm erkennt die Verzögerung bei der Replikation in einen Kinesis-Datenstrom. Im normalen Betrieb sollte `AgeOfOldestUnreplicatedRecord` nur Millisekunden betragen. Diese Zahl erhöht sich durch erfolglose Replikationsversuche, die durch die vom Kunden gesteuerte Konfiguration verursacht werden. Beispiele für kundengesteuerte Konfigurationen, die zu erfolglosen Replikationsversuchen führen, sind eine zu geringe Kapazität des Kinesis-Datenstroms, die zu einer übermäßigen Drosselung führt, oder eine manuelle Aktualisierung der Zugriffsrichtlinien des Kinesis-Datenstroms, die DynamoDB daran hindert, Daten zum Datenstrom hinzuzufügen. Um diese Metrik so niedrig wie möglich zu halten, müssen

Sie für die richtige Bereitstellung der Kinesis-Datenstromkapazität sorgen und sicherstellen, dass die Berechtigungen von DynamoDB unverändert bleiben.

Absicht: Dieser Alarm kann erfolglose Replikationsversuche und die daraus resultierende Verzögerung bei der Replikation im Kinesis-Datenstrom überwachen.

Statistik: Maximum

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Stellen Sie den Schwellenwert entsprechend der gewünschten Replikationsverzögerung ein, gemessen in Millisekunden. Dieser Wert hängt von den Anforderungen Ihres Workloads und der erwarteten Leistung ab.

Zeitraum: 300

Datenpunkte bis Alarm: 3

Auswertungszeiträume: 3

Vergleichsoperator: GREATER_THAN_THRESHOLD

FailedToReplicateRecordCount

Abmessungen:TableName, DelegatedOperation

Alarmbeschreibung: Dieser Alarm erkennt die Anzahl der Datensätze, die DynamoDB nicht in Ihren Kinesis-Datenstrom replizieren konnte. Bestimmte Elemente, die größer als 34 KB sind, können sich vergrößern, um Datensätze zu ändern, die größer als die Elementgrößengrenze von 1 MB von Kinesis Data Streams sind. Diese Größenerweiterung tritt auf, wenn diese Elemente, die größer als 34 KB sind, eine große Anzahl von booleschen oder leeren Attributwerten enthalten. Boolesche und leere Attributwerte werden in DynamoDB als 1 Byte gespeichert, erweitern sich jedoch auf bis zu 5 Byte, wenn sie mit Standard-JSON für die Kinesis-Data-Streams-Replikation serialisiert werden. DynamoDB kann solche Änderungsdatensätze nicht in Ihren Kinesis Dats Stream replizieren. DynamoDB überspringt diese Änderungsdatensätze und repliziert automatisch nachfolgende Datensätze.

Absicht: Dieser Alarm kann die Anzahl der Datensätze überwachen, die DynamoDB aufgrund der Elementgrößenbeschränkung für Kinesis-Datenströme nicht in Ihren Kinesis-Datenstrom repliziert hat.

Statistik: Summe

Empfohlener Schwellenwert: 0,0

Begründung des Schwellenwerts: Setzen Sie den Schwellenwert auf 0, um alle Datensätze zu erkennen, die DynamoDB nicht replizieren konnte.

Zeitraum: 60

Datenpunkte bis Alarm: 1

Auswertungszeiträume: 1

Vergleichsoperator: GREATER_THAN_THRESHOLD

ReadThrottleEvents

Abmessungen: TableName

Alarmbeschreibung: Dieser Alarm erkennt, ob eine hohe Anzahl von Leseanfragen für die DynamoDB-Tabelle gedrosselt wird. Informationen zur Behebung des Problems finden Sie unter Behebung von Drosselungsproblemen in Amazon DynamoDB.

Absicht: Dieser Alarm kann eine anhaltende Drosselung von Leseanfragen an die DynamoDB-Tabelle erkennen. Eine anhaltende Drosselung von Leseanfragen kann sich negativ auf Ihre Workload-Lesevorgänge auswirken und die Gesamteffizienz des Systems verringern.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Legen Sie den Schwellenwert entsprechend dem erwarteten Leseverkehr für die DynamoDB-Tabelle fest und berücksichtigen Sie dabei ein akzeptables Maß an Drosselung. Es ist wichtig, dass Sie überwachen, ob Sie zu wenig Ressourcen zur Verfügung haben und nicht durchgängig eine Drosselung verursachen. Sie können auch historische Daten analysieren, um die akzeptable Drosselungsstufe für den Anwendungs-Workload zu ermitteln, und dann den Schwellenwert so einstellen, dass er höher als die übliche Drosselungsstufe ist. Gedrosselte Anfragen sollten von der Anwendung oder dem Service erneut versucht werden, da sie vorübergehend sind. Daher kann ein sehr niedriger Schwellenwert dazu führen, dass der Alarm zu empfindlich ist, was zu unerwünschten Zustandsübergängen führen kann.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

ReadThrottleEvents

Abmessungen:TableName, GlobalSecondaryIndexName

Alarmbeschreibung: Dieser Alarm erkennt, ob eine hohe Anzahl von Leseanfragen für den Global Secondary Index der DynamoDB-Tabelle gedrosselt wird. Informationen zur Behebung des Problems finden Sie unter Behebung von Drosselungsproblemen in Amazon DynamoDB.

Absicht: Dieser Alarm kann eine anhaltende Drosselung von Leseanfragen für den Global Secondary Index der DynamoDB-Tabelle erkennen. Eine anhaltende Drosselung von Leseanfragen kann sich negativ auf Ihre Workload-Lesevorgänge auswirken und die Gesamteffizienz des Systems verringern.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Legen Sie den Schwellenwert entsprechend dem erwarteten Leseverkehr für die DynamoDB-Tabelle fest und berücksichtigen Sie dabei ein akzeptables Maß an Drosselung. Es ist wichtig, dass Sie überwachen, ob Sie unterdurchschnittlich ausgelastet sind und nicht durchgehend eine Drosselung verursachen. Sie können auch historische Daten analysieren, um eine akzeptable Drosselungsstufe für den Anwendungs-Workload zu finden, und dann den Schwellenwert so einstellen, dass er über der üblichen akzeptablen Drosselungsstufe liegt. Gedrosselte Anfragen sollten von der Anwendung oder dem Service erneut versucht werden, da sie vorübergehend sind. Daher kann ein sehr niedriger Schwellenwert dazu führen, dass der Alarm zu empfindlich ist, was zu unerwünschten Zustandsübergängen führen kann.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

ReplicationLatency

Abmessungen:TableName, ReceivingRegion

Alarmbeschreibung: Der Alarm erkennt, wenn das Replikat in einer Region für die globale Tabelle hinter der Quellregion zurückbleibt. Die Latenz kann sich erhöhen, wenn eine AWS Region herabgesetzt wird und Sie in dieser Region über eine Replikattabelle verfügen. In diesem Fall können Sie die Lese- und Schreibaktivitäten Ihrer Anwendung vorübergehend in eine andere AWS Region umleiten. Wenn Sie globale Tabellen der Version 2017.11.29 (veraltet) verwenden, sollten Sie überprüfen, ob die Schreibkapazitätseinheiten (WCUs) für jede der Replikattabellen identisch sind. Sie können auch sicherstellen, dass Sie die Empfehlungen unter Bewährte Methoden und Anforderungen für das Kapazitätsmanagement befolgen.

Absicht: Dieser Alarm kann erkennen, wenn die Replikattabelle in einer Region bei der Replikation der Änderungen aus einer anderen Region in Verzug gerät. Dies könnte dazu führen, dass Ihr Replikat von den anderen Replikaten abweicht. Es ist nützlich, die Replikationslatenz jeder AWS Region zu kennen und eine Warnung zu erhalten, wenn diese Replikationslatenz kontinuierlich zunimmt. Die Replikation der Tabelle gilt nur für globale Tabellen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Der empfohlene Schwellenwert für diesen Alarm hängt stark von Ihrem Anwendungsfall ab. Replikationslatenzen von mehr als 3 Minuten sind in der Regel ein Grund für Untersuchungen. Prüfen Sie die Wichtigkeit und die Anforderungen der Replikationsverzögerung, analysieren Sie historische Trends und wählen Sie dann den Schwellenwert entsprechend aus.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: GREATER_THAN_THRESHOLD

SuccessfulRequestLatency

Abmessungen:TableName, Bedienung

Alarmbeschreibung: Dieser Alarm erkennt eine hohe Latenz für den DynamoDB-Tabellenvorgang (angezeigt durch den Dimensionswert von `Operation` im Alarm). In diesem Dokument zur Fehlerbehebung finden Sie Informationen zur Behebung von Latenzproblemen in Amazon DynamoDB.

Absicht: Dieser Alarm kann eine hohe Latenz für den DynamoDB-Tabellenvorgang erkennen. Eine höhere Latenz für die Vorgänge kann sich negativ auf die Gesamteffizienz des Systems auswirken.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: DynamoDB bietet eine durchschnittliche Latenz im einstelligen Millisekundenbereich für Singleton-Operationen wie, usw. GetItem PutItem Sie können den Schwellenwert jedoch auf der Grundlage einer akzeptablen Latenztoleranz für die Art des Vorgangs und der Tabelle festlegen, die an dem Workload beteiligt sind. Sie können historische Daten dieser Metrik analysieren, um die übliche Latenz für den Tabellenvorgang zu ermitteln, und dann den Schwellenwert auf eine Zahl festlegen, die eine kritische Verzögerung für den Vorgang darstellt.

Zeitraum: 60

Datenpunkte bis Alarm: 10

Auswertungszeiträume: 10

Vergleichsoperator: GREATER_THAN_THRESHOLD

SystemErrors

Abmessungen: TableName

Alarmbeschreibung: Dieser Alarm erkennt eine anhaltend hohe Anzahl von Systemfehlern für die DynamoDB-Tabellenanfragen. Wenn Sie weiterhin 5XX-Fehler erhalten, öffnen Sie das AWS Service Health Dashboard, um nach Betriebsproblemen mit dem Service zu suchen. Sie können diesen Alarm verwenden, um benachrichtigt zu werden, falls ein längeres internes Serviceproblem von DynamoDB auftritt, und er hilft Ihnen, das Problem, mit dem Ihre Client-Anwendung konfrontiert ist, zu korrelieren. Weitere Informationen finden Sie unter Fehlerbehandlung für DynamoDB.

Absicht: Dieser Alarm kann anhaltende Systemfehler für die DynamoDB-Tabellenanfragen erkennen. Systemfehler deuten auf interne Servicefehler von DynamoDB hin und helfen dabei, das Problem, das der Client hat, zu korrelieren.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Legen Sie den Schwellenwert entsprechend dem zu erwartenden Datenverkehr fest und berücksichtigen Sie dabei ein akzeptables Maß an Systemfehlern. Sie können auch historische Daten analysieren, um die akzeptable Fehlerzahl für den Anwendungs-Workload zu ermitteln, und den Schwellenwert dann entsprechend anpassen. Systemfehler sollten von der Anwendung/dem Service erneut versucht werden, da sie vorübergehend sind. Daher kann ein sehr niedriger Schwellenwert dazu führen, dass der Alarm zu empfindlich ist, was zu unerwünschten Zustandsübergängen führen kann.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: GREATER_THAN_THRESHOLD

ThrottledPutRecordCount

Abmessungen:TableName, DelegatedOperation

Alarmbeschreibung: Dieser Alarm erkennt, dass die Datensätze während der Replikation von Change Data Capture zu Kinesis durch Ihren Kinesis-Datenstrom gedrosselt werden. Diese Drosselung erfolgt aufgrund unzureichender Kinesis-Datenstrom-Kapazität. Wenn eine übermäßige und regelmäßige Drosselung auftritt, müssen Sie möglicherweise die Anzahl der Kinesis-Stream-Shards proportional zum beobachteten Schreibdurchsatz Ihrer Tabelle erhöhen. Weitere Informationen zur Bestimmung der Größe eines Kinesis Data Streams finden Sie unter Bestimmen der anfänglichen Größe eines Kinesis Data Streams.

Absicht: Dieser Alarm kann die Anzahl der Datensätze überwachen, die von Ihrem Kinesis-Datenstrom wegen unzureichender Kapazität des Kinesis-Datenstroms gedrosselt wurden.

Statistik: Maximum

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Bei außergewöhnlichen Nutzungsspitzen kann es zu einer gewissen Drosselung kommen, gedrosselte Datensätze sollten jedoch so niedrig wie möglich gehalten werden, um eine höhere Replikationslatenz zu vermeiden (DynamoDB versucht erneut, gedrosselte Datensätze an den Kinesis-Datenstrom zu senden). Setzen Sie den Schwellenwert auf eine Zahl, mit der Sie regelmäßige übermäßige Drosselungen erkennen können. Sie können

auch historische Daten dieser Metrik analysieren, um die akzeptablen Drosselungsraten für die Anwendungs-Workload zu ermitteln. Stellen Sie den Schwellenwert auf einen Wert ein, den die Anwendung in Abhängigkeit von Ihrem Anwendungsfall tolerieren kann.

Zeitraum: 60

Datenpunkte bis Alarm: 10

Auswertungszeiträume: 10

Vergleichsoperator: GREATER_THAN_THRESHOLD

UserErrors

Dimensionen: Keine

Alarmbeschreibung: Dieser Alarm erkennt eine anhaltend hohe Anzahl von Benutzerfehlern bei DynamoDB-Tabellenanfragen. Sie können während des Problemzeitraums in den Protokollen der Client-Anwendungen nachsehen, warum die Anfragen ungültig sind. Sie können den HTTP-Statuscode 400 überprüfen, um zu sehen, welche Art von Fehler Sie erhalten, und entsprechende Maßnahmen ergreifen zu können. Möglicherweise müssen Sie die Anwendungslogik korrigieren, um gültige Anfragen zu erstellen.

Absicht: Dieser Alarm kann anhaltende Benutzerfehler bei den DynamoDB-Tabellenanfragen erkennen. Benutzerfehler bei angeforderten Vorgängen bedeuten, dass der Client ungültige Anfragen generiert und der Vorgang fehlschlägt.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Setzen Sie den Schwellenwert auf Null, um clientseitige Fehler zu erkennen. Oder Sie können ihn auf einen höheren Wert setzen, wenn Sie vermeiden möchten, dass der Alarm bei einer sehr geringen Anzahl von Fehlern ausgelöst wird. Entscheiden Sie auf der Grundlage Ihres Anwendungsfalls und des Datenverkehrs für die Anfragen.

Zeitraum: 60

Datenpunkte bis Alarm: 10

Auswertungszeiträume: 10

Vergleichsoperator: GREATER_THAN_THRESHOLD

WriteThrottleEvents

Abmessungen: TableName

Alarmbeschreibung: Dieser Alarm erkennt, ob eine hohe Anzahl von Schreibanfragen für die DynamoDB-Tabelle gedrosselt wird. Informationen zur Behebung des Problems finden Sie unter Behebung von Drosselungsproblemen in Amazon DynamoDB.

Absicht: Dieser Alarm kann eine anhaltende Drosselung von Schreibanfragen an die DynamoDB-Tabelle erkennen. Eine anhaltende Drosselung von Schreibanfragen kann sich negativ auf Ihre Workload-Schreibvorgänge auswirken und die Gesamteffizienz des Systems verringern.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Legen Sie den Schwellenwert entsprechend dem erwarteten Schreibverkehr für die DynamoDB-Tabelle fest und berücksichtigen Sie dabei ein akzeptables Maß an Drosselung. Es ist wichtig, dass Sie überwachen, ob Sie unterdurchschnittlich ausgelastet sind und nicht durchgehend eine Drosselung verursachen. Sie können auch historische Daten analysieren, um den akzeptablen Grad der Drosselung für den Anwendungs-Workload zu ermitteln, und dann den Schwellenwert auf einen Wert einstellen, der über der üblichen akzeptablen Drosselungsstufe liegt. Gedrosselte Anfragen sollten von der Anwendung/dem Service erneut versucht werden, da sie kurzlebig sind. Daher kann ein sehr niedriger Schwellenwert dazu führen, dass der Alarm zu empfindlich ist, was zu unerwünschten Zustandsübergängen führen kann.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

WriteThrottleEvents

Abmessungen:TableName, GlobalSecondaryIndexName

Alarmbeschreibung: Dieser Alarm erkennt, ob eine hohe Anzahl von Schreibanfragen für den Global Secondary Index der DynamoDB-Tabelle gedrosselt wird. Informationen zur Behebung des Problems finden Sie unter Behebung von Drosselungsproblemen in Amazon DynamoDB.

Absicht: Dieser Alarm kann eine anhaltende Drosselung von Schreibanfragen für den Global Secondary Index der DynamoDB-Tabelle erkennen. Eine anhaltende Drosselung von Schreibanfragen kann sich negativ auf Ihre Workload-Schreibvorgänge auswirken und die Gesamteffizienz des Systems verringern.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Legen Sie den Schwellenwert entsprechend dem erwarteten Schreibverkehr für die DynamoDB-Tabelle fest und berücksichtigen Sie dabei ein akzeptables Maß an Drosselung. Es ist wichtig, dass Sie überwachen, ob Sie unterdurchschnittlich ausgelastet sind und nicht durchgehend eine Drosselung verursachen. Sie können auch historische Daten analysieren, um die akzeptable Drosselungsstufe für den Anwendungs-Workload zu ermitteln, und dann den Schwellenwert auf einen Wert einstellen, der über der üblichen akzeptablen Drosselungsstufe liegt. Gedrosselte Anfragen sollten von der Anwendung/dem Service erneut versucht werden, da sie kurzlebig sind. Daher kann ein sehr niedriger Wert dazu führen, dass der Alarm zu empfindlich ist, was zu unerwünschten Zustandsübergängen führen kann.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

Amazon EBS

VolumeStalledioCheck

Abmessungen:VolumeId, InstanceId

Beschreibung des Alarms: Dieser Alarm hilft Ihnen, die I/O-Leistung Ihrer Amazon EBS-Volumes zu überwachen. Diese Prüfung erkennt grundlegende Probleme mit der Amazon EBS-Infrastruktur, wie Hardware- oder Softwareprobleme in den Speichersubsystemen, die den Amazon EBS-Volumes zugrunde liegen, Hardwareprobleme auf dem physischen Host, die sich auf die Erreichbarkeit der Amazon EBS-Volumes von Ihrer Amazon EC2 EC2-Instance aus auswirken, und kann Verbindungsprobleme zwischen der Instance und den Amazon EBS-Volumes aufdecken. Wenn der Stalled IO Check fehlschlägt, können Sie entweder warten, AWS

bis das Problem behoben ist, oder Sie können Maßnahmen ergreifen, z. B. das betroffene Volume austauschen oder die Instance, an die das Volume angehängt ist, beenden und neu starten. In den meisten Fällen, wenn diese Metrik fehlschlägt, diagnostiziert Amazon EBS Ihr Volume automatisch und stellt es innerhalb weniger Minuten wieder her.

Absicht: Dieser Alarm kann den Status Ihrer Amazon EBS-Volumes erkennen, um festzustellen, wann diese Volumes beeinträchtigt sind und I/O-Operationen nicht abgeschlossen werden können.

Statistik: Maximum

Empfohlener Schwellenwert: 1,0

Begründung des Schwellenwerts: Wird eine Statusprüfung nicht bestanden, ist der Wert dieser Metrik 1. Der Schwellenwert ist so eingestellt, dass sich der Alarm immer dann im ALARM-Status befindet, wenn die Statusüberprüfung fehlschlägt.

Zeitraum: 60

Datenpunkte bis Alarm: 10

Auswertungszeiträume: 10

Vergleichsoperator: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Amazon EC2

CPUUtilization

Abmessungen: InstanceId

Alarmbeschreibung: Dieser Alarm hilft bei der Überwachung der CPU-Auslastung einer EC2-Instance. Je nach Anwendung kann eine gleichbleibend hohe Auslastung normal sein. Wenn jedoch die Leistung beeinträchtigt ist und die Anwendung nicht durch Festplatten-I/O, Arbeitsspeicher oder Netzwerkressourcen eingeschränkt ist, kann eine ausgelastete CPU auf einen Ressourcenengpass oder auf Probleme mit der Anwendungsleistung hinweisen. Eine hohe CPU-Auslastung kann darauf hindeuten, dass ein Upgrade auf eine CPU-intensivere Instance erforderlich ist. Wenn die detaillierte Überwachung aktiviert ist, können Sie den Zeitraum auf 60 Sekunden statt auf 300 Sekunden ändern. Weitere Informationen finden Sie unter Aktivieren oder Deaktivieren der detaillierten Überwachung für Ihre Instances.

Absicht: Dieser Alarm wird verwendet, um eine hohe CPU-Auslastung zu erkennen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 80,0

Begründung des Schwellenwerts: In der Regel können Sie den Schwellenwert für die CPU-Auslastung auf 70 bis 80 % festlegen. Sie können diesen Wert jedoch an Ihr akzeptables Leistungsniveau und Ihre Workload-Merkmale anpassen. Bei einigen Systemen kann eine konstant hohe CPU-Auslastung normal sein und ist kein Anzeichen für ein Problem, bei anderen kann sie jedoch Anlass zur Sorge geben. Analysieren Sie historische Daten zur CPU-Auslastung, um die Auslastung zu ermitteln, herauszufinden, welche CPU-Auslastung für Ihr System akzeptabel ist, und legen Sie den Schwellenwert entsprechend fest.

Zeitraum: 300

Datenpunkte bis Alarm: 3

Auswertungszeiträume: 3

Vergleichsoperator: GREATER_THAN_THRESHOLD

StatusCheckFailed

Abmessungen: InstanceId

Alarmbeschreibung: Dieser Alarm hilft dabei, sowohl System-Statusprüfungen als auch den Instance-Status zu überwachen. Wenn eine der beiden Arten der Statusüberprüfung fehlschlägt, sollte sich dieser Alarm im ALARM-Status befinden.

Absicht: Dieser Alarm wird verwendet, um grundlegende Probleme mit Instances zu erkennen, einschließlich fehlgeschlagener System-Statusprüfungen und fehlgeschlagener Instance-Statusprüfungen.

Statistik: Maximum

Empfohlener Schwellenwert: 1,0

Begründung des Schwellenwerts: Wird eine Statusprüfung nicht bestanden, ist der Wert dieser Metrik 1. Der Schwellenwert ist so eingestellt, dass sich der Alarm immer dann im ALARM-Status befindet, wenn die Statusüberprüfung fehlschlägt.

Zeitraum: 300

Datenpunkte bis Alarm: 2

Auswertungszeiträume: 2

Vergleichsoperator: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

StatusCheckFailed_Angehängte Datenbanken

Abmessungen: InstanceId

Beschreibung des Alarms: Mit diesem Alarm können Sie überwachen, ob die an eine Instance angeschlossenen Amazon EBS-Volumes erreichbar sind und I/O-Operationen abschließen können. Bei dieser Statusüberprüfung werden grundlegende Probleme mit der Datenverarbeitungs- oder Amazon EBS-Infrastruktur erkannt, z. B. die folgenden:

- Hardware- oder Softwareprobleme in den Speichersubsystemen, die den Amazon EBS-Volumes zugrunde liegen
- Hardwareprobleme auf dem physischen Host, die sich auf die Erreichbarkeit der Amazon EBS-Volumes auswirken
- Verbindungsprobleme zwischen der Instance und Amazon EBS-Volumes

Wenn die angehängte EBS-Statusprüfung fehlschlägt, können Sie entweder warten, bis Amazon das Problem behoben hat, oder Sie können Maßnahmen ergreifen, z. B. die betroffenen Volumes austauschen oder die Instance beenden und neu starten.

Absicht: Dieser Alarm wird verwendet, um unerreichbare Amazon EBS-Volumes zu erkennen, die an eine Instance angehängt sind. Diese können zu Fehlern bei I/O-Vorgängen führen.

Statistik: Maximum

Empfohlener Schwellenwert: 1,0

Begründung des Schwellenwerts: Wird eine Statusprüfung nicht bestanden, ist der Wert dieser Metrik 1. Der Schwellenwert ist so eingestellt, dass sich der Alarm immer dann im ALARM-Status befindet, wenn die Statusüberprüfung fehlschlägt.

Zeitraum: 60

Datenpunkte bis Alarm: 10

Auswertungszeiträume: 10

Vergleichsoperator: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Amazon ElastiCache

CPUUtilization

Abmessungen:CacheClusterId, CacheNodeId

Beschreibung des Alarms: Dieser Alarm hilft bei der Überwachung der CPU-Auslastung für die gesamte ElastiCache Instance, einschließlich der Datenbank-Engine-Prozesse und anderer Prozesse, die auf der Instance ausgeführt werden. AWS Elasticache unterstützt zwei Engine-Typen: Memcached und Redis. Wenn Sie auf einem Memcached-Knoten eine hohe CPU-Auslastung erreichen, sollten Sie erwägen, Ihren Instance-Typ zu skalieren oder neue Cache-Knoten hinzuzufügen. Wenn Ihr Workload bei Redis hauptsächlich aus Leseanfragen besteht, sollten Sie in Erwägung ziehen, Ihrem Cache-Cluster mehr Lesereplikate hinzuzufügen. Wenn Ihr Workload hauptsächlich aus Schreibanfragen besteht, sollten Sie in Erwägung ziehen, weitere Shards hinzuzufügen, um den Workload auf mehr Primärknoten zu verteilen, wenn Sie im Cluster-Modus arbeiten, oder Ihren Instance-Typ hochzuskalieren, wenn Sie Redis im Nicht-Clustermodus betreiben.

Absicht: Dieser Alarm wird verwendet, um eine hohe CPU-Auslastung von Hosts zu erkennen. ElastiCache Es ist nützlich, einen umfassenden Überblick über die CPU-Auslastung in der gesamten Instance zu erhalten, einschließlich Prozessen, die nicht zur Engine gehören.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Legen Sie den Schwellenwert auf den Prozentsatz fest, der einer kritischen CPU-Auslastung für Ihre Anwendung entspricht. Bei Memcached kann die Engine bis zu num_threads Kerne verwenden. Bei Redis ist die Engine größtenteils Single-Thread-fähig, kann aber zusätzliche Kerne verwenden, falls verfügbar, um I/O zu beschleunigen. In den meisten Fällen können Sie den Schwellenwert auf etwa 90 % Ihrer verfügbaren CPU setzen. Da Redis mit nur einem Thread arbeitet, sollte der tatsächliche Schwellenwert als ein Bruchteil der Gesamtkapazität des Knotens berechnet werden.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

CurrConnections

Abmessungen:CacheClusterId, CacheNodeId

Alarmbeschreibung: Dieser Alarm erkennt eine hohe Verbindungsanzahl, was auf eine hohe Auslastung oder Leistungsprobleme hinweisen kann. Ein stetiger Anstieg von `CurrConnections` kann zur Erschöpfung der 65 000 verfügbaren Verbindungen führen. Dies kann darauf hinweisen, dass die Verbindungen auf der Anwendungsseite nicht ordnungsgemäß geschlossen und auf der Serverseite bestehen gelassen wurden. Sie sollten die Verwendung von Verbindungspooling oder Timeouts bei inaktiven Verbindungen in Betracht ziehen, um die Anzahl der Verbindungen zu dem Cluster zu begrenzen. Bei Redis sollten Sie in Betracht ziehen, tcp-keepalive auf Ihrem Cluster zu optimieren, um potenzielle tote Peers zu erkennen und zu beenden.

Absicht: Der Alarm hilft Ihnen dabei, hohe Verbindungszahlen zu erkennen, die sich auf die Leistung und Stabilität Ihres ElastiCache Clusters auswirken könnten.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Der empfohlene Schwellenwert für diesen Alarm hängt stark vom akzeptablen Verbindungsbereich für Ihren Cluster ab. Überprüfen Sie die Kapazität und die erwartete Arbeitslast Ihres ElastiCache Clusters und analysieren Sie die historischen Verbindungszahlen während der regulären Nutzung, um einen Basiswert festzulegen, und wählen Sie dann einen entsprechenden Schwellenwert aus. Denken Sie daran, dass jeder Knoten bis zu 65 000 gleichzeitige Verbindungen unterstützen kann.

Zeitraum: 60

Datenpunkte bis Alarm: 10

Auswertungszeiträume: 10

Vergleichsoperator: GREATER_THAN_THRESHOLD

DatabaseMemoryUsagePercentage

Abmessungen: CacheClusterId

Alarmbeschreibung: Dieser Alarm hilft Ihnen, die Speicherauslastung Ihres Clusters zu überwachen. Wenn Ihr `DatabaseMemoryUsagePercentage` 100 % erreicht, wird die Redis-Maxmemory-Richtlinie ausgelöst und es kann je nach der ausgewählten Richtlinie zu

Bereinigungen kommen. Wenn kein Objekt im Cache der Bereinigungsrichtlinie entspricht, schlagen Schreibvorgänge fehl. Einige Workloads erwarten oder verlassen sich auf Bereinigungen. Andernfalls müssen Sie die Speicherkapazität Ihres Clusters erhöhen. Sie können Ihren Cluster skalieren, indem Sie weitere Primärknoten hinzufügen, oder ihn hochskalieren, indem Sie einen größeren Knotentyp verwenden. Einzelheiten finden Sie unter Skalierung ElastiCache für Redis-Cluster.

Absicht: Dieser Alarm wird verwendet, um eine hohe Speicherauslastung Ihres Clusters zu erkennen, sodass Sie Fehler beim Schreiben in Ihren Cluster vermeiden können. Es ist hilfreich zu wissen, wann Sie Ihren Cluster hochskalieren müssen, wenn bei Ihrer Anwendung nicht mit Bereinigungen zu rechnen ist.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung der Schwellenwerte: Abhängig von den Speicheranforderungen Ihrer Anwendung und der Speicherkapazität Ihres ElastiCache Clusters sollten Sie den Schwellenwert auf den Prozentsatz festlegen, der die kritische Speichernutzung des Clusters widerspiegelt. Sie können historische Speichernutzungsdaten als Referenz für einen akzeptablen Schwellenwert für die Speichernutzung verwenden.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

EngineCPUUtilization

Abmessungen: CacheClusterId

Beschreibung des Alarms: Dieser Alarm hilft dabei, die CPU-Auslastung eines Redis-Engine-Threads innerhalb der ElastiCache Instanz zu überwachen. Häufige Gründe für hohe CPU-Engines sind lang andauernde Befehle, die viel CPU verbrauchen, eine hohe Anzahl von Anfragen, eine Zunahme neuer Client-Verbindungsanfragen in kurzer Zeit und viele Bereinigungen, wenn der Cache nicht über genügend Speicher für neue Daten verfügt. Sie sollten die Skalierung ElastiCache für Redis-Cluster in Betracht ziehen, indem Sie weitere Knoten hinzufügen oder Ihren Instanztyp skalieren.

Absicht: Dieser Alarm wird verwendet, um eine hohe CPU-Auslastung des Redis-Engine-Threads zu erkennen. Dies ist nützlich, wenn Sie die CPU-Auslastung der Datenbank-Engine selbst überwachen möchten.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 90,0

Begründung des Schwellenwerts: Stellen Sie den Schwellenwert auf einen Prozentsatz ein, der die kritische CPU-Auslastung der Engine für Ihre Anwendung widerspiegelt. Sie können Ihren Cluster anhand Ihrer Anwendung und des erwarteten Workloads vergleichen, um EngineCPUUtilization und Leistung als Referenz zu korrelieren, und dann den Schwellenwert entsprechend festlegen. In den meisten Fällen können Sie den Schwellenwert auf etwa 90 % Ihrer verfügbaren CPU festlegen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

ReplicationLag

Abmessungen: CacheClusterId

Beschreibung des Alarms: Dieser Alarm hilft dabei, den Replikationsstatus Ihres ElastiCache Clusters zu überwachen. Eine hohe Verzögerung bei der Replikation bedeutet, dass der Primärknoten oder das Replikat nicht mit dem Tempo der Replikation Schritt halten kann. Wenn Ihre Schreibaktivität zu hoch ist, sollten Sie erwägen, Ihren Cluster zu skalieren, indem Sie weitere Primärknoten hinzufügen, oder ihn mithilfe eines größeren Knotentyps hochskalieren. Einzelheiten finden Sie unter Skalierung ElastiCache für Redis-Cluster. Wenn Ihre Lesereplikate durch die Anzahl der Leseanfragen überlastet sind, sollten Sie erwägen, weitere Lesereplikate hinzuzufügen.

Absicht: Dieser Alarm wird verwendet, um eine Verzögerung zwischen Datenaktualisierungen auf dem Primärknoten und deren Synchronisation mit dem Replikatknoten zu erkennen. Er trägt dazu bei, die Datenkonsistenz eines Lesereplikats-Clusterknotens sicherzustellen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Legen Sie den Schwellenwert entsprechend den Anforderungen Ihrer Anwendung und den potenziellen Auswirkungen einer Verzögerung bei der Replikation fest. Sie sollten die von Ihrer Anwendung zu erwartenden Schreibraten und Netzwerkbedingungen für die akzeptable Verzögerung bei der Replikation berücksichtigen.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: GREATER_THAN_THRESHOLD

Amazon EC2 (**`AWS/ElasticGPUs`**)

GPU ConnectivityCheckFailed

Abmessungen: InstanceId GPUID

Alarmbeschreibung: Dieser Alarm hilft dabei, Verbindungsfehler zwischen der Instance und dem Elastic-Graphics-Beschleuniger zu erkennen. Elastic Graphics verwendet das Instance-Netzwerk zum Senden von OpenGL-Befehlen an eine remote verbundene Grafikkarte. Dazu wird auf ein Desktop mit einer OpenGL-Anwendung mit einem Elastic-Graphics-Beschleuniger typischerweise mithilfe von Remote-Zugrifftechnologie zugegriffen. Es ist wichtig, zwischen Leistungsproblemen aufgrund des OpenGL-Renderings und solchen aufgrund der Desktop-Fernzugriffstechnologie zu unterscheiden. Weitere Informationen zu diesem Problem finden Sie unter Untersuchen von Leistungsproblemen von Anwendungen.

Absicht: Dieser Alarm wird verwendet, um Verbindungsprobleme zwischen der Instance und dem Elastic-Graphics-Beschleuniger zu erkennen.

Statistik: Maximum

Empfohlener Schwellenwert: 0,0

Begründung des Schwellenwerts: Der Schwellenwert von 1 gibt an, dass die Konnektivität fehlgeschlagen ist.

Zeitraum: 300

Datenpunkte bis Alarm: 3

Auswertungszeiträume: 3

Vergleichsoperator: GREATER_THAN_THRESHOLD

GPU HealthCheckFailed

Abmessungen: InstanceId GPUID

Alarmbeschreibung: Dieser Alarm informiert Sie darüber, wenn der Status des Elastic-Graphics-Beschleuniger fehlerhaft ist. Wenn der Beschleuniger fehlerhaft ist, lesen Sie die Schritte zur Fehlerbehebung unter Beheben von Problemen im Status Fehlerhaft.

Absicht: Dieser Alarm wird verwendet, um zu erkennen, ob der Elastic-Graphics-Beschleuniger fehlerhaft ist.

Statistik: Maximum

Empfohlener Schwellenwert: 0,0

Begründung des Schwellenwerts: Der Schwellenwert 1 zeigt den Fehlschlag einer Statusprüfung an.

Zeitraum: 300

Datenpunkte bis Alarm: 3

Auswertungszeiträume: 3

Vergleichsoperator: GREATER_THAN_THRESHOLD

Amazon ECS

CPUReservation

Abmessungen: ClusterName

Alarmbeschreibung: Dieser Alarm hilft Ihnen, eine hohe CPU-Reservierung des ECS-Clusters zu erkennen. Eine hohe CPU-Reservierung kann darauf hindeuten, dass dem Cluster die registrierten CPUs für die Aufgabe ausgehen. Zur Fehlerbehebung können Sie mehr Kapazität hinzufügen, den Cluster skalieren oder Auto Scaling einrichten.

Absicht: Dieser Alarm wird verwendet, um festzustellen, ob die Gesamtzahl der für Aufgaben im Cluster reservierten CPU-Einheiten die Gesamtzahl der für den Cluster registrierten CPU-Einheiten erreicht. Auf diese Weise wissen Sie, wann Sie den Cluster hochskalieren müssen. Wenn die Gesamtzahl der CPU-Einheiten für den Cluster erreicht wird, kann dies dazu führen, dass die CPU für Aufgaben knapp wird. Wenn Sie die verwaltete Skalierung von EC2-Kapazitätsanbietern aktiviert haben oder Fargate mit Kapazitätsanbietern verknüpft haben, wird dieser Alarm nicht empfohlen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 90,0

Begründung des Schwellenwerts: Legen Sie den Schwellenwert für die CPU-Reservierung auf 90 % fest. Alternativ können Sie einen niedrigeren Wert basierend auf Cluster-Eigenschaften auswählen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

CPUUtilization

Abmessungen:ClusterName, ServiceName

Alarmbeschreibung: Dieser Alarm hilft Ihnen, eine hohe CPU-Auslastung des ECS-Services zu erkennen. Wenn keine laufende ECS-Bereitstellung stattfindet, kann eine maximale CPU-Auslastung auf einen Ressourcenengpass oder auf Probleme mit der Anwendungsleistung hinweisen. Zur Fehlerbehebung können Sie das CPU-Limit erhöhen.

Absicht: Dieser Alarm wird verwendet, um eine hohe CPU-Auslastung für den ECS-Service zu erkennen. Eine konstant hohe CPU-Auslastung kann auf einen Ressourcenengpass oder auf Probleme mit der Anwendungsleistung hinweisen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 90,0

Begründung des Schwellenwerts: Die Servicemetriken für die CPU-Auslastung könnten die Auslastung von 100 % überschreiten. Wir empfehlen jedoch, die Metrik auf hohe CPU-Auslastung

zu überwachen, um andere Services nicht zu beeinträchtigen. Legen Sie den Schwellenwert auf etwa 90-95 % fest. Wir empfehlen Ihnen, Ihre Aufgabendefinitionen so zu aktualisieren, dass sie die tatsächliche Nutzung widerspiegeln, um zukünftige Probleme mit anderen Services zu vermeiden.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

MemoryReservation

Abmessungen: ClusterName

Alarmbeschreibung: Dieser Alarm hilft Ihnen, eine hohe Speicherreservierung des ECS-Clusters zu erkennen. Eine hohe Speicherreservierung kann auf einen Ressourcenengpass für den Cluster hinweisen. Analysieren Sie zur Problembehandlung die Leistung der Serviceaufgabe, um festzustellen, ob die Speicherauslastung der Aufgabe optimiert werden kann. Sie können auch mehr Speicher registrieren oder Auto Scaling einrichten.

Absicht: Dieser Alarm wird verwendet, um festzustellen, ob die Gesamtzahl der für Aufgaben auf dem Cluster reservierten Speichereinheiten die Gesamtzahl der für den Cluster registrierten Speichereinheiten erreicht. Auf diese Weise können Sie wissen, wann Sie den Cluster hochskalieren müssen. Wenn die Gesamtzahl der Speichereinheiten für den Cluster erreicht ist, kann dies dazu führen, dass der Cluster keine neuen Aufgaben starten kann. Wenn Sie die verwaltete Skalierung von EC2-Kapazitätsanbietern aktiviert haben oder Fargate mit Kapazitätsanbietern verknüpft haben, wird dieser Alarm nicht empfohlen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 90,0

Begründung des Schwellenwerts: Legen Sie den Schwellenwert für die Speicherreservierung auf 90 % fest. Sie können diesen Wert auf der Grundlage der Cluster-Eigenschaften auf einen niedrigeren Wert anpassen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

HTTPCode_Target_5XX_Count

Abmessungen:ClusterName, ServiceName

Alarmbeschreibung: Dieser Alarm hilft Ihnen dabei, eine hohe serverseitige Fehlerzahl für den ECS-Service zu erkennen. Dies kann darauf hindeuten, dass Fehler vorliegen, die dazu führen, dass der Server Anfragen nicht bearbeiten kann. Um Fehler zu beheben, überprüfen Sie Ihre Anwendungsprotokolle.

Absicht: Dieser Alarm wird verwendet, um eine hohe Anzahl serverseitiger Fehler für den ECS-Service zu erkennen.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Berechnen Sie den Wert von etwa 5 % Ihres durchschnittlichen Datenverkehrs und verwenden Sie diesen Wert als Ausgangspunkt für den Schwellenwert. Sie können den durchschnittlichen Datenverkehr anhand der `RequestCount`-Metrik ermitteln. Sie können auch historische Daten analysieren, um die akzeptable Fehlerrate für den Anwendungs-Workload zu ermitteln, und dann den Schwellenwert entsprechend anpassen. Bei häufig auftretenden 5XX-Fehlern muss ein Alarm ausgelöst werden. Die Einstellung eines sehr niedrigen Schwellenwerts kann jedoch dazu führen, dass der Alarm zu empfindlich ist.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

TargetResponseTime

Abmessungen:ClusterName, ServiceName

Alarmbeschreibung: Dieser Alarm hilft Ihnen dabei, eine hohe Zielantwortzeit für ECS-Serviceanfragen zu erkennen. Dies kann darauf hindeuten, dass Probleme vorliegen, die

dazu führen, dass der Service Anfragen nicht rechtzeitig bearbeiten kann. Überprüfen Sie zur Fehlerbehebung anhand der Metrik CPUUtilization, ob dem Service die CPU-Auslastung ausgeht, oder überprüfen Sie die CPU-Auslastung anderer nachgeschalteter Services, von denen Ihr Service abhängt.

Absicht: Dieser Alarm wird verwendet, um eine hohe Zielantwortzeit für ECS-Serviceanfragen zu erkennen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Der empfohlene Schwellenwert für diesen Alarm hängt stark von Ihrem Anwendungsfall ab. Überprüfen Sie die Wichtigkeit und die Anforderungen der angestrebten Reaktionszeit des Services und analysieren Sie das historische Verhalten dieser Metrik, um sinnvolle Schwellenwerte zu ermitteln.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

Amazon ECS mit Container Insights

EphemeralStorageUtilized

Abmessungen:ClusterName, ServiceName

Beschreibung des Alarms: Mit diesem Alarm können Sie erkennen, dass der Fargate-Cluster einen hohen Anteil an flüchtigem Speicher nutzt. Wenn der flüchtige Speicher konstant hoch ist, können Sie die Nutzung des flüchtigen Speichers überprüfen und den flüchtigen Speicher erhöhen.

Absicht: Dieser Alarm wird verwendet, um eine hohe Nutzung des flüchtigen Speichers für den Fargate-Cluster zu erkennen. Eine gleichbleibend hohe Auslastung von flüchtigem Speicher kann darauf hinweisen, dass die Festplatte voll ist, was zu einem Ausfall des Containers führen kann.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Legen Sie den Schwellenwert auf etwa 90 % der Größe des flüchtigen Speichers fest. Sie können diesen Wert auf der Grundlage Ihrer akzeptablen Nutzung von flüchtigem Speicher des Fargate-Clusters anpassen. Bei einigen Systemen kann eine gleichbleibend hohe Auslastung von flüchtigem Speicher normal sein, während dies bei anderen zum Ausfall des Containers führen kann.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

RunningTaskCount

Abmessungen:ClusterName, ServiceName

Beschreibung des Alarms: Dieser Alarm hilft Ihnen dabei, eine geringe Anzahl laufender Aufgaben des ECS-Services zu erkennen. Wenn die Anzahl der laufenden Aufgaben zu niedrig ist, kann dies darauf hinweisen, dass die Anwendung die Auslastung des Services nicht bewältigen kann, was zu Leistungsproblemen führen kann. Wenn es keine laufende Aufgabe gibt, ist der Amazon-ECS-Service möglicherweise nicht verfügbar oder es liegen Bereitstellungsprobleme vor.

Absicht: Dieser Alarm wird verwendet, um festzustellen, ob die Anzahl der laufenden Aufgaben zu gering ist. Eine konstant niedrige Anzahl ausgeführter Aufgaben kann auf Probleme bei der Bereitstellung oder Leistung des ECS-Services hinweisen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 0,0

Begründung des Schwellenwerts: Sie können den Schwellenwert auf der Grundlage der Mindestanzahl laufender Aufgaben des ECS-Services anpassen. Wenn die Anzahl der laufenden Aufgaben 0 ist, ist der Amazon-ECS-Service nicht verfügbar.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: LESS_THAN_OR_EQUAL_TO_THRESHOLD

instance_filesystem_utilization

Abmessungen:InstanceId, ContainerInstanceId, ClusterName

Alarmbeschreibung: Dieser Alarm hilft Ihnen, eine hohe Dateisystem-Auslastung des ECS-Clusters zu erkennen. Wenn die Auslastung des Dateisystems konstant hoch ist, überprüfen Sie die Festplattennutzung.

Absicht: Dieser Alarm wird verwendet, um eine hohe Dateisystemauslastung für den Amazon-ECS-Cluster zu erkennen. Eine konstant hohe Dateisystemauslastung kann auf einen Ressourcenengpass oder auf Probleme mit der Anwendungsleistung hinweisen und die Ausführung neuer Aufgaben verhindern.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 90,0

Begründung des Schwellenwerts: In der Regel können Sie den Schwellenwert für die Dateisystem-Auslastung auf 90–95 % festlegen. Sie können diesen Wert auf der Grundlage der akzeptablen Dateisystemkapazität des Amazon-ECS-Clusters anpassen. Bei einigen Systemen ist eine konstant hohe Dateisystemauslastung möglicherweise normal und deutet nicht auf ein Problem hin, während sie bei anderen Anlass zur Sorge geben und zu Leistungseinbußen führen und die Ausführung neuer Aufgaben verhindern kann.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

Amazon EFS

PercentIOLimit

Abmessungen: FileSystemId

Alarmbeschreibung: Dieser Alarm trägt dazu bei, dass der Workload innerhalb des für das Dateisystem verfügbaren I/O-Limits bleibt. Wenn die Metrik regelmäßig das I/O-Limit erreicht,

sollten Sie in Erwägung ziehen, die Anwendung auf ein Dateisystem zu verschieben, das als Modus die maximale I/O-Leistung verwendet. Überprüfen Sie zur Fehlerbehebung die Clients, die mit dem Dateisystem verbunden sind, und die Anwendungen der Clients, die das Dateisystem drosseln.

Absicht: Dieser Alarm wird verwendet, um festzustellen, wie nah das Dateisystem an der I/O-Grenze des General-Purpose-Performance-Modus ist. Ein gleichbleibend hoher I/O-Prozentsatz kann ein Hinweis darauf sein, dass das Dateisystem in Bezug auf I/O-Anfragen nicht ausreichend skaliert werden kann und dass das Dateisystem zu einem Ressourcenengpass für die Anwendungen, die das Dateisystem verwenden, führen kann.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 100,0

Begründung des Schwellenwerts: Wenn das Dateisystem sein I/O-Limit erreicht, reagiert es möglicherweise langsamer auf Lese- und Schreibanfragen. Daher wird empfohlen, die Metrik zu überwachen, um zu vermeiden, dass sich dies auf Anwendungen auswirkt, die das Dateisystem verwenden. Der Schwellenwert kann auf etwa 100 % festgelegt werden. Dieser Wert kann jedoch auf der Grundlage der Dateisystemeigenschaften auf einen niedrigeren Wert angepasst werden.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

BurstCreditBalance

Abmessungen: FileSystemId

Alarmbeschreibung: Mit diesem Alarm kann sichergestellt werden, dass für die Nutzung des Dateisystems ein ausreichendes Guthaben verfügbar ist. Wenn kein Sptzenwert-Guthaben verfügbar ist, wird der Zugriff von Anwendungen auf das Dateisystem aufgrund niedrigen Durchsatzes eingeschränkt. Wenn die Metrik kontinuierlich auf 0 sinkt, sollten Sie erwägen, den Durchsatzmodus in den Durchsatzmodus Elastic oder Provisioned zu ändern.

Absicht: Dieser Alarm wird verwendet, um einen niedrigen Guthabenstand des Dateisystems zu erkennen. Ein gleichbleibend niedriger Spitzenwert-Guthabenstand kann ein Indikator für eine Verlangsamung des Durchsatzes und eine Zunahme der I/O-Latenz sein.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 0,0

Begründung der Schwellenwerte: Wenn das Dateisystem keine Burst-Credits mehr hat und auch wenn die Basisdurchsatzrate niedriger ist, stellt EFS weiterhin einen gemessenen Durchsatz von 1 für alle Dateisysteme MiBps bereit. Es wird jedoch empfohlen, die Metrik im Hinblick auf ein niedriges Spitzenwert-Guthaben zu überwachen, um zu verhindern, dass das Dateisystem zu einem Ressourcenengpass für die Anwendungen wird. Der Schwellenwert kann auf etwa 0 Byte festgelegt werden.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: LESS_THAN_OR_EQUAL_TO_THRESHOLD

Amazon EKS mit Container Insights

node_cpu_utilization

Abmessungen: ClusterName

Alarmbeschreibung: Dieser Alarm hilft dabei, eine hohe CPU-Auslastung in Worker-Knoten des EKS-Clusters zu erkennen. Wenn die Auslastung konstant hoch ist, kann dies darauf hindeuten, dass Sie Ihre Worker-Knoten durch Instances mit mehr CPU ersetzen müssen oder dass das System horizontal skaliert werden muss.

Absicht: Dieser Alarm hilft dabei, die CPU-Auslastung der Worker-Knoten im EKS-Cluster zu überwachen, sodass die Systemleistung nicht beeinträchtigt wird.

Statistik: Maximum

Empfohlener Schwellenwert: 80,0

Begründung des Schwellenwerts: Es wird empfohlen, den Schwellenwert auf weniger als oder gleich 80 % festzulegen, um genügend Zeit für das Debuggen des Problems zu haben, bevor das System Auswirkungen bemerkt.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

node_filesystem_utilization

Abmessungen: ClusterName

Alarmbeschreibung: Dieser Alarm hilft dabei, eine hohe Dateisystemauslastung in den Worker-Knoten des EKS-Clusters zu erkennen. Wenn die Auslastung konstant hoch ist, müssen Sie möglicherweise Ihre Worker-Knoten aktualisieren, um über ein größeres Festplattenvolumen zu verfügen, oder Sie müssen möglicherweise horizontal skalieren.

Absicht: Dieser Alarm hilft bei der Überwachung der Dateisystemauslastung der Worker-Knoten im EKS-Cluster. Wenn die Auslastung 100 % erreicht, kann dies zu einem Ausfall der Anwendung, zu Engpässen bei der Festplatten-I/O, zur Pod-Bereinigung oder dazu führen, dass der Knoten nicht mehr reagiert.

Statistik: Maximum

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Bei ausreichendem Festplattendruck (was bedeutet, dass die Festplatte voll wird), werden Knoten als nicht fehlerfrei markiert und die Pods werden aus dem Knoten entfernt. Pods auf einem Knoten mit hoher Festplattenauslastung werden entfernt, wenn das verfügbare Dateisystem unter den im Kubelet festgelegten Schwellenwerten für die Bereinigung liegt. Stellen Sie den Alarmschwellenwert so ein, dass Sie genügend Zeit haben, um zu reagieren, bevor der Knoten aus dem Cluster entfernt wird.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

node_memory_utilization

Abmessungen: ClusterName

Alarmbeschreibung: Dieser Alarm hilft bei der Erkennung einer hohen Speicherauslastung in Worker-Knoten des EKS-Clusters. Wenn die Auslastung konstant hoch ist, deutet dies möglicherweise darauf hin, dass Sie die Anzahl der Pod-Replikate skalieren oder Ihre Anwendung optimieren müssen.

Absicht: Dieser Alarm hilft dabei, die Speicherauslastung der Worker-Knoten im EKS-Cluster zu überwachen, sodass die Systemleistung nicht beeinträchtigt wird.

Statistik: Maximum

Empfohlener Schwellenwert: 80,0

Begründung des Schwellenwerts: Es wird empfohlen, den Schwellenwert auf weniger als oder gleich 80 % festzulegen, damit genügend Zeit für das Debuggen des Problems zur Verfügung steht, bevor das System Auswirkungen bemerkt.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

pod_cpu_utilization_over_pod_limit

Abmessungen:ClusterName, Namespace, Service

Alarmbeschreibung: Dieser Alarm hilft bei der Erkennung einer hohen CPU-Auslastung in Pods des EKS-Clusters. Wenn die Auslastung konstant hoch ist, deutet dies möglicherweise darauf hin, dass das CPU-Limit für den betroffenen Pod erhöht werden muss.

Absicht: Dieser Alarm hilft dabei, die CPU-Auslastung der Pods zu überwachen, die zu einem Kubernetes-Service im EKS-Cluster gehören, sodass Sie schnell erkennen können, ob der Pod eines Services mehr CPU verbraucht als erwartet.

Statistik: Maximum

Empfohlener Schwellenwert: 80,0

Begründung des Schwellenwerts: Es wird empfohlen, den Schwellenwert auf weniger als oder gleich 80 % festzulegen, damit genügend Zeit für das Debuggen des Problems zur Verfügung steht, bevor das System Auswirkungen bemerkt.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

pod_memory_utilization_over_pod_limit

Dimensionen:ClusterName, Namespace, Service

Alarmbeschreibung: Dieser Alarm hilft bei der Erkennung einer hohen Speicherauslastung in Pods des EKS-Clusters. Wenn die Auslastung konstant hoch ist, deutet dies möglicherweise darauf hin, dass das Speicherlimit für den betroffenen Pod erhöht werden muss.

Absicht: Dieser Alarm hilft dabei, die Speicherauslastung der Pods im EKS-Cluster zu überwachen, sodass die Systemleistung nicht beeinträchtigt wird.

Statistik: Maximum

Empfohlener Schwellenwert: 80,0

Begründung des Schwellenwerts: Es wird empfohlen, den Schwellenwert auf weniger als oder gleich 80 % festzulegen, damit genügend Zeit für das Debuggen des Problems zur Verfügung steht, bevor das System Auswirkungen bemerkt.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

Amazon Kinesis Data Streams

GetRecords.IteratorAgeMilliseconds

Abmessungen: StreamName

Alarmbeschreibung: Dieser Alarm kann erkennen, ob das Höchstalter des Iterators zu hoch ist. Bei Datenverarbeitungsanwendungen in Echtzeit sollten Sie die Datenaufbewahrung

entsprechend der Verzögerungstoleranz konfigurieren. Dies ist normalerweise innerhalb von Minuten der Fall. Verwenden Sie bei Anwendungen, die historische Daten verarbeiten, diese Metrik, um die Aufholgeschwindigkeit zu überwachen. Eine schnelle Lösung, um Datenverlust zu verhindern, besteht darin, die Aufbewahrungsdauer zu verlängern, während Sie das Problem beheben. Sie können auch die Anzahl der Worker erhöhen, die Datensätze in Ihrer Verbraucheranwendung verarbeiten. Die häufigsten Ursachen für einen allmählichen Anstieg des Iterator-Alters sind unzureichende physische Ressourcen oder eine Logik für die Datensatzverarbeitung, die nicht mit einem Anstieg des Stream-Durchsatzes skaliert. Weitere Einzelheiten finden Sie unter dem Link.

Absicht: Dieser Alarm wird verwendet, um zu erkennen, ob die Daten in Ihrem Stream ablaufen, weil sie zu lange aufbewahrt werden oder weil die Verarbeitung der Datensätze zu langsam ist. Er hilft Ihnen, Datenverluste zu vermeiden, nachdem Sie 100 % der Stream-Aufbewahrungszeit erreicht haben.

Statistik: Maximum

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Der empfohlene Schwellenwert für diesen Alarm hängt stark von der Aufbewahrungsdauer des Streams und der Toleranz der Verarbeitungsverzögerungen für die Datensätze ab. Überprüfen Sie Ihre Anforderungen und analysieren Sie historische Trends. Legen Sie dann den Schwellenwert auf die Anzahl von Millisekunden fest, die einer kritischen Verarbeitungsverzögerung entspricht. Wenn das Alter eines Iterators 50 % des Aufbewahrungszeitraums (standardmäßig 24 Stunden, anpassbar auf bis zu 365 Tage) überschreitet, besteht die Gefahr des Datenverlusts durch Ablauf des Datensatzes. Sie können die Metrik überwachen, um sicherzustellen, dass keiner Ihrer Shards jemals dieses Limit erreicht.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: GREATER_THAN_THRESHOLD

GetRecords. Erfolg

Abmessungen: StreamName

Alarmbeschreibung: Diese Metrik wird immer dann erhöht, wenn Ihre Verbraucher erfolgreich Daten aus Ihrem Stream lesen. `GetRecords` gibt keine Daten zurück, wenn eine Ausnahme

auslöst wird. Die häufigste Ausnahme ist `ProvisionedThroughputExceededException`, weil die Anfragerate für den Stream zu hoch ist oder weil der verfügbare Durchsatz für die angegebene Sekunde bereits bereitgestellt wurde. Reduzieren Sie die Häufigkeit oder den Umfang Ihrer Anfragen. Weitere Informationen finden Sie unter Streams Limits im Entwicklerhandbuch von Amazon Kinesis Data Streams und unter Error Retries and Exponential Backoff in AWS.

Absicht: Dieser Alarm kann erkennen, ob das Abrufen von Datensätzen aus dem Stream durch Verbraucher fehlschlägt. Wenn Sie einen Alarm für diese Metrik einrichten, können Sie proaktiv alle Probleme mit dem Datenverbrauch erkennen, z. B. erhöhte Fehlerraten oder einen Rückgang erfolgreicher Abrufe. Auf diese Weise können Sie rechtzeitig Maßnahmen ergreifen, um potenzielle Probleme zu lösen und eine reibungslose Datenverarbeitungspipeline aufrechtzuerhalten.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Je nachdem, wie wichtig es ist, Datensätze aus dem Stream abzurufen, legen Sie den Schwellenwert auf der Grundlage der Toleranz Ihrer Anwendung für fehlgeschlagene Datensätze fest. Der Schwellenwert sollte dem entsprechenden Prozentsatz erfolgreicher Vorgänge entsprechen. Sie können historische GetRecords Metrikdaten als Referenz für die akzeptable Ausfallrate verwenden. Sie sollten bei der Festlegung des Schwellenwerts auch Wiederholungsversuche in Betracht ziehen, da fehlgeschlagene Datensätze wiederholt werden können. Auf diese Weise wird verhindert, dass vorübergehende Spitzenwerte unnötige Warnmeldungen auslösen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: LESS_THAN_THRESHOLD

PutRecord. Erfolg

Abmessungen: StreamName

Alarmbeschreibung: Dieser Alarm erkennt, wenn die Anzahl der fehlgeschlagenen `PutRecord`-Vorgänge den Schwellenwert überschreitet. Untersuchen Sie die Datenproduzent-Protokolle, um die Hauptursachen der Fehler zu ermitteln. Der häufigste Grund ist ein unzureichender

bereitgestellter Durchsatz auf dem Shard, der `ProvisionedThroughputExceededException` verursacht hat. Dies geschieht, weil die Anfragerate für den Stream zu hoch ist oder der Durchsatz, der in den Shard aufgenommen werden sollte, zu hoch ist. Reduzieren Sie die Häufigkeit oder den Umfang Ihrer Anfragen. Weitere Informationen finden Sie unter Streams Limits and Error Retries und Exponential Backoff in. AWS

Absicht: Dieser Alarm kann erkennen, ob die Erfassung von Datensätzen in den Stream fehlschlägt. Er hilft Ihnen dabei, Probleme beim Schreiben von Daten in den Stream zu identifizieren. Indem Sie bei dieser Metrik einen Alarm einrichten, können Sie proaktiv alle Probleme erkennen, die Produzenten bei der Veröffentlichung von Daten im Stream haben, z. B. erhöhte Fehlerquoten oder einen Rückgang erfolgreicher Veröffentlichungen von Datensätzen. Auf diese Weise können Sie rechtzeitig Maßnahmen ergreifen, um potenzielle Probleme zu beheben und einen zuverlässigen Datenerfassungsprozess aufrechtzuerhalten.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Je nachdem, wie wichtig die Datenerfassung und -verarbeitung für Ihren Service ist, legen Sie den Schwellenwert auf der Grundlage der Toleranz Ihrer Anwendung für fehlerhafte Datensätze fest. Der Schwellenwert sollte dem entsprechenden Prozentsatz erfolgreicher Vorgänge entsprechen. Sie können historische PutRecord Metrikdaten als Referenz für die akzeptable Ausfallrate verwenden. Sie sollten bei der Festlegung des Schwellenwerts auch Wiederholungsversuche in Betracht ziehen, da fehlgeschlagene Datensätze wiederholt werden können.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: LESS_THAN_THRESHOLD

PutRecords.FailedRecords

Abmessungen: StreamName

Alarmbeschreibung: Dieser Alarm erkennt, wenn die Anzahl der fehlgeschlagenen `PutRecords` den Schwellenwert überschreitet. Kinesis Data Streams versucht, alle Datensätze in jeder

`PutRecords`-Anfrage zu verarbeiten, aber der Ausfall eines einzelnen Datensatzes verhindert nicht die Verarbeitung nachfolgender Datensätze. Der Hauptgrund für diese Fehler ist die Überschreitung des Durchsatzes eines Streams oder eines einzelnen Shards. Häufige Ursachen sind Verkehrsspitzen und Netzwerklatenzen, die dazu führen, dass Datensätze ungleichmäßig im Stream ankommen. Sie sollten erfolglos verarbeitete Datensätze erkennen und sie in einem späteren Aufruf erneut versuchen. Weitere Informationen finden Sie unter Behandlung von Fehlern bei der Verwendung PutRecords.

Absicht: Dieser Alarm kann konsistente Fehler erkennen, wenn Datensätze mithilfe eines Batch-Vorgangs in Ihren Stream aufgenommen werden. Wenn Sie einen Alarm für diese Metrik einrichten, können Sie proaktiv eine Zunahme fehlgeschlagener Datensätze erkennen und so rechtzeitig Maßnahmen ergreifen, um die zugrunde liegenden Probleme zu beheben und einen reibungslosen und zuverlässigen Datenerfassungsprozess sicherzustellen.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Legen Sie den Schwellenwert auf die Anzahl der fehlgeschlagenen Datensätze fest, was der Toleranz der Anwendung gegenüber fehlerhaften Datensätzen entspricht. Sie können historische Daten als Referenz für den akzeptablen Fehlerwert verwenden. Sie sollten bei der Festlegung des Schwellenwerts auch Wiederholungsversuche in Betracht ziehen, da fehlgeschlagene Datensätze bei nachfolgenden PutRecords Aufrufen erneut versucht werden können.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

ReadProvisionedThroughputExceeded

Abmessungen: StreamName

Alarmbeschreibung: Der Alarm verfolgt die Anzahl der Datensätze, die zu einer Drosselung der Lesedurchsatzkapazität führen. Wenn Sie feststellen, dass Sie ständig gedrosselt werden, sollten Sie erwägen, Ihrem Stream weitere Shards hinzuzufügen, um den bereitgestellten Lesedurchsatz zu erhöhen. Wenn mehr als eine Verbraucheranwendung auf dem Stream läuft

und diese Anwendungen das `GetRecords`-Limit gemeinsam nutzen, empfehlen wir Ihnen, neue Verbraucheranwendungen über Enhanced Fan-Out zu registrieren. Wenn das Hinzufügen weiterer Shards die Anzahl der Drosselungen nicht verringert, haben Sie möglicherweise einen „heißen“ Shard, der von mehr Shards gelesen wird als andere Shards. Aktivieren Sie die erweiterte Überwachung, suchen Sie den „heißen“ Shard und teilen Sie ihn auf.

Absicht: Dieser Alarm kann erkennen, ob Verbraucher gedrosselt werden, wenn sie Ihren bereitgestellten Lesedurchsatz überschreiten (bestimmt durch die Anzahl Ihrer Shards). In diesem Fall können Sie nicht aus dem Stream lesen, und der Stream kann mit der Sicherung beginnen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Normalerweise können gedrosselte Anfragen erneut versucht werden. Wenn der Schwellenwert auf Null gesetzt wird, ist der Alarm daher zu empfindlich. Eine konsequente Drosselung kann sich jedoch auf das Lesen aus dem Stream auswirken und sollte den Alarm auslösen. Legen Sie den Schwellenwert auf einen Prozentsatz fest, der den gedrosselten Anfragen für die Anwendung entspricht, und versuchen Sie es erneut mit den Konfigurationen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

SubscribeToShardEvent.MillisBehindLatest

Abmessungen:StreamName, ConsumerName

Alarmbeschreibung: Dieser Alarm erkennt, wenn die Verzögerung der Datensatzverarbeitung in der Anwendung den Schwellenwert überschreitet. Vorübergehende Probleme wie API-Betriebsausfälle bei einer nachgelagerten Anwendung können zu einem plötzlichen Anstieg in der Metrik führen. Sie sollten untersuchen, ob sie regelmäßig auftreten. Eine häufige Ursache ist, dass der Verbraucher Datensätze nicht schnell genug verarbeitet, weil nicht genügend physische Ressourcen zur Verfügung stehen oder die Logik der Datensatzverarbeitung nicht mit einem Anstieg des Stream-Durchsatzes skaliert wurde. Das Blockieren von Aufrufen im kritischen Pfad ist häufig die Ursache für eine Verlangsamung der Datensatzverarbeitung. Sie können Ihre

Parallelität erhöhen, indem Sie die Anzahl der Shards erhöhen. Sie sollten auch sicherstellen, dass die zugrunde liegenden Verarbeitungsknoten bei hoher Nachfrage über ausreichende physische Ressourcen verfügen.

Absicht: Dieser Alarm kann eine Verzögerung beim Ereignis „Abonnement des Shards“ für den Stream erkennen. Dies deutet auf eine Verarbeitungsverzögerung hin und kann dazu beitragen, potenzielle Probleme mit der Leistung der Verbraucheranwendung oder dem allgemeinen Zustand des Streams zu identifizieren. Wenn die Verarbeitungsverzögerung erheblich ansteigt, sollten Sie etwaige Engpässe oder Ineffizienzen bei Verbraucheranwendungen untersuchen und beheben, um die Datenverarbeitung in Echtzeit sicherzustellen und Datenrückstände zu minimieren.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Der empfohlene Schwellenwert für diesen Alarm hängt stark von der Verzögerung ab, die Ihre Anwendung tolerieren kann. Prüfen Sie die Anforderungen Ihrer Anwendung, analysieren Sie historische Trends und wählen Sie dann einen entsprechenden Schwellenwert aus. Wenn der SubscribeToShard Anruf erfolgreich ist, empfängt Ihr Kunde bis zu 5 Minuten lang SubscribeToShardEvent Ereignisse über die persistente Verbindung. Danach müssen Sie SubscribeToShard erneut anrufen, um das Abonnement zu verlängern, wenn Sie weiterhin Aufzeichnungen erhalten möchten.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

WriteProvisionedThroughputExceeded

Abmessungen: StreamName

Alarmbeschreibung: Dieser Alarm erkennt, wenn die Anzahl der Datensätze, die zu einer Drosselung der Schreibdurchsatzkapazität geführt haben, den Schwellenwert erreicht hat. Wenn Ihre Produzenten Ihren bereitgestellten Schreibdurchsatz (bestimmt durch die Anzahl Ihrer Shards) überschreiten, werden sie gedrosselt und Sie können keine Datensätze in den Stream aufnehmen. Um einer ständigen Drosselung entgegenzuwirken, sollten Sie erwägen, Ihrem Stream Shards hinzuzufügen. Dies erhöht den bereitgestellten Schreibdurchsatz und verhindert

zukünftige Drosselungen. Sie sollten bei der Aufnahme von Datensätzen auch die Wahl des Partitionsschlüssels berücksichtigen. Ein zufälliger Partitionsschlüssel wird bevorzugt, da er die Datensätze, wann immer möglich, gleichmäßig auf die Shards des Streams verteilt.

Absicht: Dieser Alarm kann erkennen, ob Ihre Produzenten aufgrund einer Drosselung des Streams oder Shards für das Schreiben von Datensätzen abgewiesen werden. Wenn sich Ihr Stream im Bereitstellungsmodus befindet, können Sie durch die Einstellung dieses Alarms proaktiv Maßnahmen ergreifen, wenn der Datenstrom seine Grenzen erreicht. So können Sie die bereitgestellte Kapazität optimieren oder geeignete Skalierungsmaßnahmen ergreifen, um Datenverlust zu vermeiden und eine reibungslose Datenverarbeitung aufrechtzuerhalten.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Normalerweise können gedrosselte Anfragen erneut versucht werden. Wenn Sie den Schwellenwert also auf Null setzen, wird der Alarm zu empfindlich. Eine konsistente Drosselung kann sich jedoch auf das Schreiben in den Stream auswirken. Sie sollten den Alarmschwellenwert so einstellen, dass dies erkannt wird. Legen Sie den Schwellenwert auf einen Prozentsatz fest, der den gedrosselten Anfragen für die Anwendung entspricht, und versuchen Sie es erneut mit den Konfigurationen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

Lambda

ClaimedAccountConcurrency

Dimensionen: Keine

Beschreibung des Alarms: Mit diesem Alarm können Sie überwachen, ob sich die Parallelität Ihrer Lambda-Funktionen dem Parallelitätslimit Ihres Kontos auf Regionsebene nähert. Eine Funktion wird gedrosselt, wenn sie das Gleichzeitigkeitslimit erreicht. Sie können folgende Aktionen durchführen, um Drosselung zu vermeiden.

1. Fordern Sie eine Erhöhung der Parallelität in dieser Region an.
2. Identifizieren und reduzieren Sie jegliche ungenutzte reservierte Parallelität oder bereitgestellte Parallelität.
3. Identifizieren Sie Leistungsprobleme bei den Funktionen, um die Verarbeitungsgeschwindigkeit und damit den Durchsatz zu verbessern.
4. Erhöhen Sie die Batchgröße der Funktionen, sodass bei jedem Funktionsaufruf mehr Nachrichten verarbeitet werden.

Absicht: Dieser Alarm kann proaktiv erkennen, ob sich die Parallelität Ihrer Lambda-Funktionen dem Parallelitätskontingent auf Regionsebene Ihres Kontos nähert, sodass Sie entsprechend handeln können. Funktionen werden gedrosselt, wenn das Kontingent für Parallelität auf Regionsebene für das Konto `ClaimedAccountConcurrency` erreicht wird. Wenn Sie Reserved Concurrency (RC) oder Provisioned Concurrency (PC) verwenden, erhalten Sie mit diesem Alarm einen besseren Überblick über die Parallelitätsnutzung als bei einem Alarm. `ConcurrentExecutions`

Statistik: Maximum

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Sie sollten den Wert von etwa 90% der Quote für Parallelität berechnen, die für das Konto in der Region festgelegt wurde, und das Ergebnis als Schwellenwert verwenden. Standardmäßig verfügt Ihr Konto über ein Gleichzeitigkeitskontingent von 1 000 für alle Funktionen in einer Region. Sie sollten jedoch das Kontingent Ihres Kontos im Service-Kontingents-Dashboard überprüfen.

Zeitraum: 60

Datenpunkte bis Alarm: 10

Auswertungszeiträume: 10

Vergleichsoperator: GREATER_THAN_THRESHOLD

Fehler

Abmessungen: FunctionName

Alarmbeschreibung: Dieser Alarm erkennt hohe Fehlerzahlen. Zu den Fehlern gehören die vom Code ausgelösten Ausnahmen sowie die von der Lambda-Laufzeit ausgelösten Ausnahmen.

Sie können die mit der Funktion verbundenen Protokolle überprüfen, um das Problem zu diagnostizieren.

Absicht: Dieser Alarm hilft dabei, hohe Fehlerzahlen bei Funktionsaufrufen zu erkennen.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Setzen Sie den Schwellenwert auf eine Zahl größer als Null. Der genaue Wert kann von der Fehlertoleranz in Ihrer Anwendung abhängen. Machen Sie sich mit der Wichtigkeit der Aufrufe vertraut, die die Funktion verarbeitet. Bei einigen Anwendungen kann jeder Fehler inakzeptabel sein, während andere Anwendungen eine gewisse Fehlerquote zulassen können.

Zeitraum: 60

Datenpunkte bis Alarm: 3

Auswertungszeiträume: 3

Vergleichsoperator: GREATER_THAN_THRESHOLD

Drosselungen

Abmessungen: FunctionName

Alarmbeschreibung: Dieser Alarm erkennt eine hohe Anzahl gedrosselter Aufrufanfragen. Drosselung findet statt, wenn keine Gleichzeitigkeit für das Hochskalieren verfügbar ist. Es gibt mehrere Möglichkeiten, um dieses Problem zu beheben. 1) Fordern Sie beim AWS Support in dieser Region eine Erhöhung der Parallelität an. 2) Identifizieren Sie Leistungsprobleme in der Funktion, um die Verarbeitungsgeschwindigkeit und damit den Durchsatz zu verbessern. 3) Erhöhen Sie die Batchgröße der Funktion, sodass bei jedem Funktionsaufruf mehr Nachrichten verarbeitet werden.

Absicht: Dieser Alarm hilft dabei, eine hohe Anzahl gedrosselter Aufrufanfragen für eine Lambda-Funktion zu erkennen. Es ist wichtig zu wissen, ob Anfragen aufgrund von Drosselung ständig abgelehnt werden und ob Sie die Leistung der Lambda-Funktionen verbessern oder die Gleichzeitigkeitskapazität erhöhen müssen, um eine ständige Drosselung zu vermeiden.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Setzen Sie den Schwellenwert auf eine Zahl größer als Null. Der genaue Wert des Schwellenwerts kann von der Toleranz der Anwendung abhängen. Stellen Sie den Schwellenwert entsprechend den Nutzungs- und Skalierungsanforderungen der Funktion ein.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Duration (Dauer)

Abmessungen: FunctionName

Alarmbeschreibung: Dieser Alarm erkennt lange Verarbeitungszeiten eines Ereignisses durch eine Lambda-Funktion. Lange Zeiträume können auf Änderungen im Funktionscode zurückzuführen sein, die dazu führen, dass die Ausführung der Funktion länger dauert, oder dass die Abhängigkeiten der Funktion länger dauern.

Absicht: Dieser Alarm kann eine lange Laufzeitdauer einer Lambda-Funktion erkennen. Eine hohe Laufzeitdauer weist darauf hin, dass der Aufruf einer Funktion länger dauert, und kann sich auch auf die Gleichzeitigkeitskapazität des Aufrufs auswirken, wenn Lambda eine höhere Anzahl von Ereignissen verarbeitet. Es ist wichtig zu wissen, ob die Ausführung der Lambda-Funktion ständig länger dauert als erwartet.

Statistik: p90

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Der Schwellenwert für die Dauer hängt von Ihrer Anwendung und Ihren Workloads sowie Ihren Leistungsanforderungen ab. Legen Sie für Hochleistungsanforderungen den Schwellenwert auf einen kürzeren Zeitraum fest, um festzustellen, ob die Funktion die Erwartungen erfüllt. Sie können auch historische Daten für die Dauer analysieren, um festzustellen, ob die benötigte Zeit den Leistungserwartungen der Funktion entspricht, und dann den Schwellenwert auf einen längeren Zeitraum als den historischen Durchschnitt festlegen. Stellen Sie sicher, dass Sie den Schwellenwert unter dem konfigurierten Funktions-Timeout einstellen.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: GREATER_THAN_THRESHOLD

ConcurrentExecutions

Abmessungen: FunctionName

Alarmbeschreibung: Mit diesem Alarm können Sie überwachen, ob sich die Gleichzeitigkeit der Funktion dem Gleichzeitigkeitslimit Ihres Kontos auf Regionsebene nähert. Eine Funktion wird gedrosselt, wenn sie das Gleichzeitigkeitslimit erreicht. Sie können folgende Aktionen durchführen, um Drosselung zu vermeiden.

1. Beantragen Sie eine Erhöhung der Parallelität in dieser Region.
2. Identifizieren Sie Leistungsprobleme in den Funktionen, um die Verarbeitungsgeschwindigkeit und damit den Durchsatz zu verbessern.
3. Erhöhen Sie die Batchgröße der Funktionen, sodass bei jedem Funktionsaufruf mehr Nachrichten verarbeitet werden.

Um einen besseren Überblick über reservierte Parallelität und bereitgestellte Parallelität zu erhalten, sollten Sie stattdessen einen Alarm für die neue Metrik einrichten. `ClaimedAccountConcurrency`

Absicht: Dieser Alarm kann proaktiv erkennen, ob sich die Gleichzeitigkeit der Funktion dem Gleichzeitigkeitskontingent auf Regionsebene Ihres Kontos nähert, sodass Sie entsprechend handeln können. Eine Funktion wird gedrosselt, wenn sie das Gleichzeitigkeitskontingent auf Regionsebene des Kontos erreicht.

Statistik: Maximum

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Legen Sie den Schwellenwert auf etwa 90 % des Gleichzeitigkeitskontingents fest, das für das Konto in der Region festgelegt wurde. Standardmäßig verfügt Ihr Konto über ein Gleichzeitigkeitskontingent von 1 000 für alle Funktionen in einer Region. Sie können jedoch das Kontingent Ihres Kontos überprüfen, da es erhöht werden kann, indem Sie sich an den Support wenden. AWS

Zeitraum: 60

Datenpunkte bis Alarm: 10

Auswertungszeiträume: 10

Vergleichsoperator: GREATER_THAN_THRESHOLD

Lambda Insights

Wir empfehlen, für die folgenden Lambda-Insights-Metriken Alarme nach bewährten Methoden einzustellen.

memory_utilization

Dimensionen: function_name

Alarmbeschreibung: Dieser Alarm wird verwendet, um festzustellen, ob sich die Speicherauslastung einer Lambda-Funktion dem konfigurierten Grenzwert nähert. Zur Fehlerbehebung können Sie versuchen, 1) Ihren Code optimieren. 2) Die richtige Größe Ihrer Speicherzuweisung festlegen, indem Sie den Speicherbedarf genau abschätzen. Sie können dafür in Lambda Power Tuning mehr dazu erfahren. 3) Verwenden von Verbindungspooling. Informationen zum Verbindungspooling für die RDS-Datenbank finden Sie unter Amazon-RDS-Proxy mit Lambda verwenden. 4) Sie können auch erwägen, Ihre Funktionen so zu gestalten, dass zwischen Aufrufen keine großen Datenmengen im Speicher gespeichert werden.

Absicht: Dieser Alarm wird verwendet, um zu erkennen, ob sich die Speicherauslastung für die Lambda-Funktion dem konfigurierten Grenzwert nähert.

Statistik: Durchschnitt

Schwellenwertvorschlag: 90,0

Begründung des Schwellenwerts: Stellen Sie den Schwellenwert auf 90 % ein, um eine Warnung zu erhalten, wenn die Speicherauslastung 90 % des zugewiesenen Speichers überschreitet. Sie können diesen Wert auf einen niedrigeren Wert anpassen, wenn Sie Bedenken hinsichtlich der Speicherauslastung des Workloads haben. Sie können auch die historischen Daten für diese Metrik überprüfen und den Schwellenwert entsprechend festlegen.

Zeitraum: 60

Datenpunkte bis Alarm: 10

Auswertungszeiträume: 10

ComparisonOperator: GREATER_THAN_THRESHOLD

Amazon VPC (**`AWS/NATGateway`**)

ErrorPortAllocation

Abmessungen: NatGatewayId

Alarmbeschreibung: Dieser Alarm hilft zu erkennen, wenn NAT-Gateway neuen Verbindungen keine Ports zuweisen kann. Informationen zur Behebung dieses Problems finden Sie unter Beheben von Portzuweisungsfehlern auf NAT-Gateway.

Absicht: Dieser Alarm wird verwendet, um festzustellen, ob NAT-Gateway keinen Quell-Port zuordnen konnte.

Statistik: Summe

Empfohlener Schwellenwert: 0,0

Begründung des Schwellenwerts: Wenn der Wert von größer als Null ErrorPortAllocation ist, bedeutet das, dass zu viele gleichzeitige Verbindungen zu einem einzigen beliebten Ziel über NatGateway geöffnet sind.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: GREATER_THAN_THRESHOLD

PacketsDropCount

Abmessungen: NatGatewayId

Alarmbeschreibung: Dieser Alarm hilft zu erkennen, wenn Pakete von NAT-Gateway verworfen werden. Dies kann auf ein Problem mit NAT Gateway zurückzuführen sein. Überprüfen Sie

daher das AWS Service Health Dashboard, um den Status von AWS NAT Gateway in Ihrer Region zu überprüfen. Dies kann Ihnen helfen, das Netzwerkproblem im Zusammenhang mit dem Datenverkehr über NAT-Gateway zu korrelieren.

Absicht: Dieser Alarm wird verwendet, um festzustellen, ob Pakete von NAT-Gateway verworfen werden.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Sie sollten den Wert von 0,01 Prozent des Gesamtdatenverkehrs auf NAT-Gateway berechnen und dieses Ergebnis als Schwellenwert verwenden. Verwenden Sie historische Daten des Datenverkehrs auf NAT-Gateway, um den Schwellenwert zu bestimmen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

AWS Privater Link (**`AWS/PrivateLinkEndpoints`**)

PacketsDropped

Dimensionen: VPC-ID, VPC-Endpunkt-ID, Endpunkt-Typ, Subnetz-ID, Servicename

Alarmbeschreibung: Dieser Alarm hilft zu erkennen, ob der Endpunkt oder der Endpunkt-Service fehlerhaft ist, indem er die Anzahl der vom Endpunkt verworfenen Pakete überwacht. Beachten Sie, dass Pakete, die größer als 8 500 Byte sind und am VPC-Endpunkt ankommen, verworfen werden. Informationen zur Fehlerbehebung finden Sie unter Verbindungsprobleme zwischen einem Schnittstellen-VPC-Endpunkt und einem Endpunkt-Service.

Absicht: Dieser Alarm wird verwendet, um festzustellen, ob der Endpunkt oder Endpunkt-Service in einem fehlerhaften Zustand ist.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Legen Sie den Schwellenwert entsprechend dem Anwendungsfall fest. Wenn Sie über den fehlerhaften Status des Endpunkts oder Endpunkt-Services informiert werden möchten, sollten Sie den Schwellenwert niedrig ansetzen, damit Sie die Chance haben, das Problem zu beheben, bevor ein großer Datenverlust entsteht. Sie können historische Daten verwenden, um die Toleranz gegenüber verworfenen Paketen zu ermitteln und den Schwellenwert entsprechend festzulegen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

AWS Privater Link (**`AWS/PrivateLinkServices`**)

RstPacketsSent

Dimensionen: Service Id, Load Balancer Arn, Az

Alarmbeschreibung: Dieser Alarm hilft Ihnen dabei, fehlerhafte Ziele eines Endpunkt-Services anhand der Anzahl der an Endpunkte gesendeten Reset-Pakete zu erkennen. Wenn Sie Verbindungsfehler mit einem Benutzer Ihres Dienstes debuggen, können Sie anhand der RstPacketsSent Metrik überprüfen, ob der Dienst Verbindungen zurücksetzt oder ob etwas anderes auf dem Netzwerkpfad fehlschlägt.

Absicht: Dieser Alarm wird verwendet, um fehlerhafte Ziele eines Endpunkt-Services zu erkennen.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Der Schwellenwert hängt vom Anwendungsfall ab. Wenn Ihr Anwendungsfall toleriert, dass Ziele fehlerhaft sind, können Sie den Schwellenwert hoch setzen. Wenn der Anwendungsfall fehlerhafte Ziele nicht toleriert, können Sie den Schwellenwert als sehr niedrig festlegen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

Amazon RDS

CPUUtilization

Abmessungen: DB InstanceIdentifier

Beschreibung des Alarms: Dieser Alarm hilft bei der Überwachung einer gleichbleibend hohen CPU-Auslastung. Bei der CPU-Auslastung wird die Betriebszeit gemessen. Erwägen Sie, Erweiterte Überwachung oder Performance Insights zu verwenden, um für MariaDB, MySQL, Oracle und PostgreSQL zu überprüfen, welche Wartezeit die meiste CPU-Zeit (`guest`, `irq`, `wait`, `nice`, usw.) beansprucht. Analysieren Sie dann, welche Abfragen die größte Menge an CPU-Leistung verbrauchen. Wenn Sie Ihre Workload nicht optimieren können, sollten Sie erwägen, zu einer größeren DB-Instance-Klasse zu wechseln.

Absicht: Dieser Alarm wird verwendet, um eine konstant hohe CPU-Auslastung zu erkennen, um sehr hohe Reaktionszeiten und Timeouts zu vermeiden. Wenn Sie die CPU-Auslastung im Mikro-Bursting überprüfen möchten, können Sie eine kürzere Zeit für die Auswertung von Alarmen festlegen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 90,0

Begründung des Schwellenwerts: Zufällige CPU-Auslastungsspitzen beeinträchtigen möglicherweise nicht die Datenbankleistung, aber ein anhaltend hoher CPU-Wert kann bevorstehende Datenbankanfragen behindern. Abhängig von der Gesamt-Workload der Datenbank kann eine hohe CPU-Auslastung Ihrer RDS/Aurora-Instance die Gesamtleistung beeinträchtigen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

DatabaseConnections

Abmessungen: DB InstanceIdentifier

Beschreibung des Alarms: Dieser Alarm erkennt eine hohe Anzahl von Verbindungen. Überprüfen Sie die bestehenden Verbindungen und beenden Sie alle Verbindungen, die sich im Ruhemodus befinden oder die nicht ordnungsgemäß geschlossen wurden. Erwägen Sie die Verwendung von Verbindungspooling, um die Anzahl neuer Verbindungen zu begrenzen. Alternativ können Sie die DB-Instance-Größe erhöhen, um eine Klasse mit mehr Speicher und damit einem höheren Standardwert für `max_connections` zu verwenden, oder erhöhen Sie den Wert von `max_connections` in RDS und Aurora MySQL und PostgreSQL für die aktuelle Klasse, wenn sie Ihre Workload unterstützen kann.

Absicht: Dieser Alarm wird verwendet, um zu verhindern, dass Verbindungen abgewiesen werden, wenn die maximale Anzahl von DB-Verbindungen erreicht ist. Dieser Alarm wird nicht empfohlen, wenn Sie die DB-Instance-Klasse häufig ändern, da sich dadurch der Arbeitsspeicher und die standardmäßige maximale Anzahl von Verbindungen ändern.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Die Anzahl der zulässigen Verbindungen hängt von der Größe Ihrer DB-Instance-Klasse und den Datenbank-Engine-spezifischen Parametern in Bezug auf Prozesse/Verbindungen ab. Sie sollten einen Wert zwischen 90 und 95 % der maximalen Anzahl von Verbindungen für Ihre Datenbank berechnen und dieses Ergebnis als Schwellenwert verwenden.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

EBS% ByteBalance

Abmessungen: DB InstanceIdentifier

Beschreibung des Alarms: Dieser Alarm hilft dabei, einen niedrigen Prozentsatz der verbleibenden Durchsatz-Credits zu überwachen. Informationen zur Fehlerbehebung finden Sie unter Latenzprobleme in RDS.

Absicht: Dieser Alarm wird verwendet, um einen niedrigen Prozentsatz an verbleibenden Durchsatz-Credits im Burst-Bucket zu erkennen. Ein niedriger Byte-Restprozentsatz kann zu Durchsatzengpässen führen. Dieser Alarm wird für Aurora-PostgreSQL-Instances nicht empfohlen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 10,0

Begründung des Schwellenwerts: Ein Durchsatzguthaben unter 10 % wird als schlecht angesehen, und Sie sollten den Schwellenwert entsprechend festlegen. Sie können auch einen niedrigeren Schwellenwert festlegen, wenn Ihre Anwendung einen niedrigeren Durchsatz für den Workload toleriert.

Zeitraum: 60

Datenpunkte bis Alarm: 3

Auswertungszeiträume: 3

Vergleichsoperator: LESS_THAN_THRESHOLD

EBSIOBalance%

Abmessungen: DB InstanceIdentifier

Beschreibung des Alarms: Dieser Alarm hilft dabei, einen niedrigen Prozentsatz der verbleibenden IOPS-Credits zu überwachen. Informationen zur Fehlerbehebung finden Sie unter Latenzprobleme in RDS.

Absicht: Dieser Alarm wird verwendet, um einen niedrigen Prozentsatz an verbleibenden I/O-Credits im Burst-Bucket zu erkennen. Ein niedriger IOPS-Restprozentsatz kann zu IOPS-Engpässen führen. Dieser Alarm wird für Aurora-Instances nicht empfohlen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 10,0

Begründung des Schwellenwerts: Ein IOPS-Guthaben unter 10 % wird als schlecht angesehen, und Sie sollten den Schwellenwert entsprechend festlegen. Sie können auch einen niedrigeren Schwellenwert festlegen, wenn Ihre Anwendung einen niedrigeren IOPS für den Workload toleriert.

Zeitraum: 60

Datenpunkte bis Alarm: 3

Auswertungszeiträume: 3

Vergleichsoperator: LESS_THAN_THRESHOLD

FreeableMemory

Abmessungen: DB InstanceIdentifier

Beschreibung des Alarms: Dieser Alarm hilft bei der Überwachung eines niedrigen freisetzbarem Speicherplatzes. Dies kann bedeuten, dass die Datenbankverbindungen stark ansteigen oder dass Ihre Instance unter hohem Speicherdruck steht. Prüfen Sie den Speicherdruck, indem Sie die CloudWatch Messwerte für `SwapUsage` `zusätzlich zu `FreeableMemory` überwachen. Wenn der Speicherverbrauch der Instance häufig zu hoch ist, bedeutet dies, dass Sie die Workload prüfen sollten oder die Instance aktualisieren müssen. Erwägen Sie für Aurora-Reader-DB-Instance, dem Cluster zusätzliche Reader-DB-Instances hinzuzufügen. Weitere Informationen zur Fehlerbehebung von Aurora finden Sie unter Probleme mit freisetzbarem Speicher.

Absicht: Dieser Alarm wird verwendet, um zu verhindern, dass nicht genügend Arbeitsspeicher zur Verfügung steht, was zu abgelehnten Verbindungen führen kann.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Je nach Workload und Instance-Klasse können unterschiedliche Werte für den Schwellenwert angemessen sein. Idealerweise sollte der verfügbare Arbeitsspeicher über längere Zeiträume nicht unter 25 % des gesamten Arbeitsspeichers fallen. Für Aurora können Sie diesen Schwellenwert nahe an 5 % einstellen, denn wenn sich diese Metrik dem Wert 0 nähert, ist die DB-Instance so weit wie möglich hochskaliert. Sie können das historische Verhalten dieser Metrik analysieren, um sinnvolle Schwellenwerte zu ermitteln.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: LESS_THAN_THRESHOLD

FreeLocalStorage

Abmessungen: DB InstanceIdentifier

Beschreibung des Alarms: Dieser Alarm hilft bei der Überwachung auf niedrigen freien lokalen Speicher. Aurora PostgreSQL-Compatible Edition verwendet lokalen Speicher zum Speichern von Fehlerprotokollen und temporären Dateien. Aurora MySQL verwendet lokalen Speicher zum Speichern von Fehlerprotokollen, allgemeinen Protokollen, langsamen Abfrageprotokollen, Prüfungsprotokollen und temporären Tabellen, die nicht von InnoDB stammen. Diese lokalen Speicher-Volumes werden von Amazon EBS Store gestützt und können durch Einsatz einer größeren DB-Instance-Klasse erweitert werden. Informationen zur Fehlerbehebung finden Sie unter Aurora PostgreSQL-kompatibel und MySQL-kompatibel.

Absicht: Dieser Alarm wird verwendet, um zu erkennen, wie nahe die Aurora-DB-Instance am Erreichen des lokalen Speicherlimits ist, wenn Sie nicht Aurora Serverless v2 oder höher verwenden. Lokaler Speicher kann seine Kapazität erreichen, wenn Sie nicht persistente Daten, wie temporäre Tabellen- und Protokolldateien, im lokalen Speicher speichern. Dieser Alarm kann einen out-of-space Fehler verhindern, der auftritt, wenn Ihrer DB-Instance der lokale Speicher ausgeht.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Sie sollten etwa 10–20 % der verfügbaren Speichermenge auf der Grundlage der Geschwindigkeit und des Trends der Volumennutzung berechnen und dieses Ergebnis dann als Schwellenwert verwenden, um proaktiv Maßnahmen zu ergreifen, bevor das Volumen sein Limit erreicht.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: LESS_THAN_THRESHOLD

FreeStorageSpace

Abmessungen: DB InstanceIdentifier

Beschreibung des Alarms: Dieser Alarm sucht nach einer geringen Menge an verfügbarem Speicherplatz. Erwägen Sie eine Skalierung Ihres Datenbankspeichers, wenn Sie häufig an Speicherkapazitätsgrenzen stoßen. Planen Sie einen gewissen Puffer ein, um unvorhergesehene Nachfragesteigerungen seitens Ihrer Anwendungen bewältigen zu können. Erwägen Sie alternativ, das Auto Scaling des RDS-Speichers zu aktivieren. Erwägen Sie außerdem, mehr Speicherplatz freizugeben, indem Sie ungenutzte oder veraltete Daten und Protokolle löschen. Weitere Informationen finden Sie im Dokument zur RDS-Speicherauslastung und im Dokument zu PostgreSQL-Speicherproblemen.

Absicht: Dieser Alarm trägt dazu bei, Probleme mit vollem Speicherplatz zu vermeiden. Dies kann Ausfallzeiten vermeiden, wenn der DB-Instance der Speicherplatz ausgeht. Wir empfehlen, diesen Alarm nicht zu verwenden, wenn Sie Auto Scaling für Speicher aktiviert haben oder wenn Sie die Speicherkapazität der Datenbank-Instance häufig ändern.

Statistik: Minimum

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Der Schwellenwert hängt vom aktuell zugewiesenen Speicherplatz ab. In der Regel sollten Sie den Wert von 10 % des zugewiesenen Speicherplatzes berechnen und dieses Ergebnis als Schwellenwert verwenden.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: LESS_THAN_THRESHOLD

MaximumUsedTransactionAusweise

Abmessungen: DB InstanceIdentifier

Beschreibung des Alarms: Dieser Alarm hilft dabei, den Transaktions-ID-Wraparound für PostgreSQL zu verhindern. Lesen Sie die Schritte zur Fehlerbehebung in diesem Blog, um das Problem zu untersuchen und zu beheben. In diesem Blog können Sie sich auch näher mit den Konzepten, häufigen Problemen und bewährten Methoden von Autovacuum vertraut machen.

Absicht: Dieser Alarm wird verwendet, um den Transaktions-ID-Wraparound für PostgreSQL zu verhindern.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 1,0E9

Begründung des Schwellenwerts: Wenn Sie diesen Schwellenwert auf 1 Milliarde festlegen, sollten Sie Zeit haben, das Problem zu untersuchen. Der Standardwert für autovacuum_freeze_max_age ist 200 Millionen. Wenn das Alter der ältesten Transaktion 1 Milliarde beträgt, hat autovacuum ein Problem damit, diesen Schwellenwert unter dem Ziel von 200 Millionen Transaktions-IDs zu halten.

Zeitraum: 60

Datenpunkte bis Alarm: 1

Auswertungszeiträume: 1

Vergleichsoperator: GREATER_THAN_THRESHOLD

ReadLatency

Abmessungen: DB InstanceIdentifier

Beschreibung des Alarms: Dieser Alarm hilft bei der Überwachung einer hohen Leselatenz. Wenn die Speicherlatenz hoch ist, liegt das daran, dass die Workload die Ressourcenlimits überschreitet. Sie können die I/O-Auslastung im Verhältnis zur Instance und der zugewiesenen Speicherkonfiguration überprüfen. Informationen finden Sie unter Fehlerbehebung der Latenz von Amazon-EBS-Volumes, die durch einen IOPS-Engpass verursacht wurden. Für Aurora können Sie zu einer Instance-Klasse mit I/O-optimierter Speicherkonfiguration wechseln. Anleitungen finden Sie unter I/O in Aurora planen.

Absicht: Dieser Alarm wird verwendet, um eine hohe Leselatenz zu erkennen. Datenbankfestplatten haben normalerweise eine geringe Lese-/Schreiblatenz, sie können jedoch Probleme haben, die zu Vorgängen mit hoher Latenz führen können.

Statistik: p90

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Der empfohlene Schwellenwert für diesen Alarm hängt stark von Ihrem Anwendungsfall ab. Leselatenzen von mehr als 20 Millisekunden sind wahrscheinlich

ein Grund für Untersuchungen. Sie können auch einen höheren Schwellenwert festlegen, wenn Ihre Anwendung eine höhere Latenz für Lesevorgänge haben kann. Überprüfen Sie die Wichtigkeit und die Anforderungen der Leselatenz und analysieren Sie das historische Verhalten dieser Metrik, um sinnvolle Schwellenwerte zu ermitteln.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

ReplicaLag

Abmessungen: DB InstanceIdentifier

Beschreibung des Alarms: Dieser Alarm hilft Ihnen, die Anzahl von Sekunden zu Verfolgen, die ein Replikat hinter der primären Instance liegt. Ein PostgreSQL-Lesereplikat meldet eine Replikationsverzögerung von bis zu fünf Minuten, wenn keine Benutzertransaktionen in der Quell-Datenbank-Instance erfolgen. Wenn die ReplicaLag Metrik 0 erreicht, hat das Replikat die primäre DB-Instance erreicht. Wenn die ReplicaLag Metrik -1 zurückgibt, ist die Replikation derzeit nicht aktiv. Anleitungen zu RDS PostgreSQL finden Sie unter Bewährte Methoden für die Replikation. Informationen zur Problembehandlung `ReplicaLag` und damit verbundenen Fehlern finden Sie unter Problembehandlung. ReplicaLag

Absicht: Dieser Alarm kann die Verzögerung bei der Replikation erkennen, die den Datenverlust widerspiegelt, der bei einem Ausfall der primären Instance auftreten könnte. Wenn das Replikat zu weit hinter der primären Instance zurückbleibt und die primäre Instance ausfällt, fehlen dem Replikat Daten, die sich in der primären Instance befanden.

Statistik: Maximum

Empfohlener Schwellenwert: 60,0

Begründung des Schwellenwerts: In der Regel hängt die akzeptable Verzögerung von der Anwendung ab. Wir empfehlen nicht mehr als 60 Sekunden.

Zeitraum: 60

Datenpunkte bis Alarm: 10

Auswertungszeiträume: 10

Vergleichsoperator: GREATER_THAN_THRESHOLD

WriteLatency

Abmessungen: DB InstanceIdentifier

Beschreibung des Alarms: Dieser Alarm hilft bei der Überwachung einer hohen Schreiblatenz. Wenn die Speicherlatenz hoch ist, liegt das daran, dass die Workload die Ressourcenlimits überschreitet. Sie können die I/O-Auslastung im Verhältnis zur Instance und der zugewiesenen Speicherkonfiguration überprüfen. Informationen finden Sie unter Fehlerbehebung der Latenz von Amazon-EBS-Volumes, die durch einen IOPS-Engpass verursacht wurden. Für Aurora können Sie zu einer Instance-Klasse mit I/O-optimierter Speicherkonfiguration wechseln. Anleitungen finden Sie unter I/O in Aurora planen.

Absicht: Dieser Alarm wird verwendet, um eine hohe Schreiblatenz zu erkennen. Obwohl Datenbankfestplatten in der Regel eine geringe Lese-/Schreiblatenz aufweisen, kann es bei ihnen zu Problemen kommen, die zu Vorgängen mit hoher Latenz führen. Wenn Sie dies überwachen, können Sie sicherstellen, dass die Festplattenlatenz so gering wie erwartet ist.

Statistik: p90

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Der empfohlene Schwellenwert für diesen Alarm hängt stark von Ihrem Anwendungsfall ab. Schreiblatenzen von mehr als 20 Millisekunden sind wahrscheinlich ein Grund für Untersuchungen. Sie können auch einen höheren Schwellenwert festlegen, wenn Ihre Anwendung eine höhere Latenz für Schreibvorgänge haben kann. Überprüfen Sie die Wichtigkeit und die Anforderungen der Schreiblatenz und analysieren Sie das historische Verhalten dieser Metrik, um sinnvolle Schwellenwerte zu ermitteln.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

DBLoad

Abmessungen: DB InstanceIdentifier

Beschreibung des Alarms: Dieser Alarm hilft bei der Überwachung einer hohen DB-Latenz. Wenn die Anzahl der Prozesse die Anzahl der vCPUs übersteigt, werden die Prozesse in die Warteschlange gestellt. Wenn die Warteschlange länger wird, wird die Leistung beeinträchtigt. Wenn die DB-Last häufig über der maximalen vCPU liegt und der primäre Wartezustand „CPU“ lautet, ist die CPU überlastet. In diesem Fall können Sie `CPUUtilization`, `DBLoadCPU` und Aufgaben in der Warteschlange für Performance Insights/Enhanced Monitoring überwachen. Sie sollten Verbindungen zur Instance drosseln, SQL-Abfragen mit hoher CPU-Last anpassen oder eine größere Instance-Klasse in Betracht ziehen. Hohe und konsistente Instances von Wartezuständen deuten darauf hin, dass es möglicherweise Engpässe oder Probleme mit Ressourcenkonflikten gibt, die behoben werden müssen.

Absicht: Dieser Alarm wird verwendet, um eine hohe DB-Auslastung zu erkennen. Eine hohe DB-Auslastung kann zu Leistungsproblemen in der DB-Instance führen. Dieser Alarm gilt nicht für Serverless-DB-Instances.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Der maximale vCPU-Wert wird anhand der Anzahl der vCPU (virtuellen CPU)-Cores für Ihre DB-Instance bestimmt. Abhängig von der maximalen vCPU können unterschiedliche Werte für den Schwellenwert angemessen sein. Idealerweise sollte die DB-Auslastung nicht über die vCPU-Linie hinausgehen.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: GREATER_THAN_THRESHOLD

AuroraVolumeBytesLeftTotal

Abmessungen: DB ClusterIdentifier

Beschreibung des Alarms: Dieser Alarm hilft bei der Überwachung eines niedrigen verbleibenden Gesamtvolumens. Wenn das verbleibende Gesamtvolumen die Größenbeschränkung erreicht, meldet der Cluster einen out-of-space Fehler. Der Aurora-Speicher wird automatisch mit den Daten im Cluster-Volume skaliert und je nach DB-Engine-Version auf bis zu 128 TiB oder 64 TiB erweitert. So können Sie Speicherkosten senken, indem Sie Tabellen und Datenbanken

verwerfen, die Sie nicht mehr benötigen. Weitere Informationen finden Sie unter Skalierung des Speichers.

Absicht: Dieser Alarm wird verwendet, um zu erkennen, wie nahe der Aurora-Cluster an der maximalen Volumengröße ist. Dieser Alarm kann einen out-of-space Fehler verhindern, der auftritt, wenn auf Ihrem Cluster nicht mehr genügend Speicherplatz zur Verfügung steht. Dieser Alarm wird nur für Aurora MySQL empfohlen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Sie sollten etwa 10–20 % der tatsächlichen maximalen Größe auf der Grundlage der Geschwindigkeit und des Trends der Volumennutzung berechnen und dieses Ergebnis dann als Schwellenwert verwenden, um proaktiv Maßnahmen zu ergreifen, bevor das Volumen sein Limit erreicht.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: LESS_THAN_THRESHOLD

AuroraBinlogReplicaLag

Dimensionen: DBClusterIdentifier, role=Writer

Beschreibung des Alarms: Dieser Alarm hilft dabei, den Fehlerstatus der Aurora-Writer-Instancereplikation zu überwachen. Weitere Informationen finden Sie unter AWS Regionale Replikation von Aurora MySQL-DB-Clustern. Informationen zur Fehlerbehebung finden Sie unter Probleme mit der Aurora-MySQL-Replikation.

Absicht: Dieser Alarm wird verwendet, um festzustellen, ob sich die Writer-Instance in einem Fehlerstatus befindet und die Quelle nicht replizieren kann. Dieser Alarm wird nur für Aurora MySQL empfohlen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: -1,0

Begründung des Schwellenwerts: Wir empfehlen, -1 als Schwellenwert zu verwenden, da Aurora MySQL diesen Wert veröffentlicht, wenn sich das Replikat in einem Fehlerstatus befindet.

Zeitraum: 60

Datenpunkte bis Alarm: 2

Auswertungszeiträume: 2

Vergleichsoperator: LESS_THAN_OR_EQUAL_TO_THRESHOLD

BlockedTransactions

Abmessungen: DB InstanceIdentifier

Beschreibung des Alarms: Dieser Alarm hilft bei der Überwachung einer hohen Anzahl blockierter Transaktionen in einer Aurora-DB-Instance. Blockierte Transaktionen können entweder mit einem Rollback oder einem Commit enden. Hohe Parallelität, ungenutzte Transaktionen oder lang andauernde Transaktionen können zu blockierten Transaktionen führen. Informationen zur Fehlerbehebung finden Sie in der Aurora-MySQL-Dokumentation.

Absicht: Dieser Alarm wird verwendet, um eine hohe Anzahl blockierter Transaktionen in einer Aurora-DB-Instance zu erkennen, um Transaktions-Rollbacks und Leistungseinbußen zu verhindern.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Sie sollten 5 % aller Transaktionen Ihrer Instance anhand der `ActiveTransactions`-Metrik berechnen und dieses Ergebnis als Schwellenwert verwenden. Sie können auch die Wichtigkeit und die Anforderungen von blockierten Transaktionen überprüfen und das historische Verhalten dieser Metrik analysieren, um sinnvolle Schwellenwerte zu ermitteln.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

BufferCacheHitRatio

Abmessungen: DB InstanceIdentifier

Beschreibung des Alarms: Dieser Alarm hilft Ihnen dabei, eine gleichbleibend niedrige Cache-Trefferquote des Aurora-Clusters zu überwachen. Wenn die Trefferrate niedrig ist, zeigt dies an, dass Ihre Abfragen für diese DB-Instance häufig zum Datenträger geleitet werden. Untersuchen Sie zur Fehlerbehebung die Workload, um herauszufinden, welche Abfragen dieses Verhalten verursachen und wenden Sie sich an das Dokument Empfehlungen für DB-Instance-RAM.

Absicht: Dieser Alarm wird verwendet, um eine konstant niedrige Cache-Trefferquote zu erkennen, um einen anhaltenden Leistungsabfall in der Aurora-Instance zu verhindern.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 80,0

Begründung des Schwellenwerts: Sie können den Schwellenwert für die Trefferquote im Puffercache auf 80 % festlegen. Sie können diesen Wert jedoch an Ihr akzeptables Leistungsniveau und Ihre Workload-Merkmale anpassen.

Zeitraum: 60

Datenpunkte bis Alarm: 10

Auswertungszeiträume: 10

Vergleichsoperator: LESS_THAN_THRESHOLD

EngineUptime

Abmessungen: DBClusterIdentifier, role=Writer

Beschreibung des Alarms: Dieser Alarm hilft dabei, die geringe Ausfallzeit der Writer-DB-Instance zu überwachen. Die Writer-DB-Instance kann aufgrund eines Neustarts, einer Wartung, eines Upgrades oder eines Failovers ausfallen. Wenn die Betriebszeit wegen eines Failovers im Cluster 0 erreicht, und der Cluster über ein oder mehrere Aurora-Replikate verfügt, dann wird während eines Failover-Ereignisses ein Aurora-Replikat zur primären Writer-Instance hochgestuft. Erwägen Sie, ein oder mehrere Aurora-Replikate in zwei oder mehreren verschiedenen Availability Zones zu erstellen, um die Verfügbarkeit Ihres DB-Clusters zu erhöhen. Weitere Informationen finden Sie unter Faktoren, die die Ausfallzeit von Aurora beeinflussen.

Absicht: Dieser Alarm wird verwendet, um festzustellen, ob die Aurora-Writer-DB-Instance ausgefallen ist. Dadurch kann ein lang andauernder Ausfall der Writer-Instance verhindert werden, der aufgrund eines Absturzes oder Failovers auftritt.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 0,0

Begründung des Schwellenwerts: Ein Fehlerereignis hat eine kurze Unterbrechung zufolge, während der die Lese- und Schreibvorgänge mit einer Ausnahme fehlschlagen. Jedoch wird der Service im Normalfall in weniger als 60 Sekunden und oft sogar schon nach 30 Sekunden wiederhergestellt.

Zeitraum: 60

Datenpunkte bis Alarm: 2

Auswertungszeiträume: 2

Vergleichsoperator: LESS_THAN_OR_EQUAL_TO_THRESHOLD

RollbackSegmentHistoryListLength

Abmessungen: DB InstanceIdentifier

Beschreibung des Alarms: Dieser Alarm hilft dabei, eine konstant hohe Länge des Rollback-Segmentverlaufs einer Aurora-Instance zu überwachen. Wenn die Länge der InnoDB-Verlaufsliste zu groß wird, was auf eine große Anzahl alter Zeilenversionen hinweist, werden Abfragen und Datenbank-Abschaltungen langsamer. Weitere Informationen und Schritte zur Problembehebung finden Sie in der Dokumentation zur InnoDB-Verlaufsliste, die erheblich erweitert wurde.

Absicht: Dieser Alarm wird verwendet, um eine konstant hohe Länge des Rollback-Segmentverlaufs zu erkennen. Dies kann Ihnen helfen, anhaltende Leistungseinbußen und eine hohe CPU-Auslastung in der Aurora-Instance zu verhindern. Dieser Alarm wird nur für Aurora MySQL empfohlen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 1 000 000,0

Begründung des Schwellenwerts: Wenn Sie diesen Schwellenwert auf 1 Million festlegen, sollten Sie Zeit haben, das Problem zu untersuchen. Sie können diesen Wert jedoch an Ihr akzeptables Leistungsniveau und Ihre Workload-Merkmale anpassen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

StorageNetworkThroughput

Abmessungen: DBClusterIdentifier, role=Writer

Beschreibung des Alarms: Dieser Alarm hilft bei der Überwachung eines hohen Durchsatzes im Speichernetzwerk. Wenn der Durchsatz im Speichernetzwerk die gesamte Netzwerkbandbreite der EC2-Instance übersteigt, kann dies zu einer hohen Lese- und Schreiblatenz führen, was zu Leistungseinbußen führen kann. Sie können Ihren EC2-Instance-Typ von der Konsole aus überprüfen. AWS Überprüfen Sie zur Fehlerbehebung alle Änderungen an den Schreib- und Leselatenzen und prüfen Sie, ob Sie auch bei dieser Metrik einen Alarm ausgelöst haben. Wenn das der Fall ist, überprüfen Sie Ihr Workload-Muster zu den Zeiten, in denen der Alarm ausgelöst wurde. Auf diese Weise können Sie herausfinden, ob Sie Ihre Workload optimieren können, um den gesamten Netzwerk-Datenverkehr zu reduzieren. Wenn dies nicht möglich ist, müssen Sie möglicherweise eine Skalierung Ihrer Instance in Betracht ziehen.

Absicht: Dieser Alarm wird verwendet, um einen hohen Durchsatz im Speichernetzwerk zu erkennen. Durch die Erkennung eines hohen Durchsatzes können Netzwerkpaketverluste und Leistungseinbußen verhindert werden.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Sie sollten etwa 80–90 % der gesamten Netzwerkbandbreite des EC2-Instance-Typs berechnen und dieses Ergebnis dann als Schwellenwert verwenden, um proaktiv Maßnahmen zu ergreifen, bevor die Netzwerkpakete betroffen sind. Sie können auch die Wichtigkeit und die Anforderungen des Speichernetzwerk-Durchsatzes überprüfen und das historische Verhalten dieser Metrik analysieren, um sinnvolle Schwellenwerte zu ermitteln.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

Amazon Route 53 Public Data Plane

HealthCheckStatus

Abmessungen: HealthCheckId

Alarmbeschreibung: Dieser Alarm hilft dabei, fehlerhafte Endgeräte gemäß den Zustandsprüfern zu erkennen. Um den Grund für einen Fehler zu ermitteln, der zu einem fehlerhaften Status führt, verwenden Sie die Registerkarte Route 53 Health Check Console, um den Status für jede Region sowie den letzten Fehler der Zustandsprüfung anzuzeigen. Auf der Registerkarte Status wird auch der Grund angezeigt, warum der Endpunkt als fehlerhaft gemeldet wurde. Weitere Informationen finden Sie unter Schritte zur Fehlerbehebung.

Absicht: Bei diesem Alarm werden Route53-Zustandsprüfer verwendet, um fehlerhafte Endpunkte zu erkennen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 1,0

Begründung des Schwellenwerts: Der Status des Endpunkts wird als 1 gemeldet, wenn er fehlerfrei ist. Alles, was kleiner als 1 ist, zählt als fehlerhaft.

Zeitraum: 60

Datenpunkte bis Alarm: 3

Auswertungszeiträume: 3

Vergleichsoperator: LESS_THAN_THRESHOLD

Amazon S3

4xxErrors

Abmessungen:BucketName, FilterId

Alarmbeschreibung: Dieser Alarm hilft uns dabei, die Gesamtzahl der 4XX-Fehlerstatuscodes zu melden, die als Antwort auf Kundenanfragen ausgegeben wurden. 403-Fehlercodes können

auf eine falsche IAM-Richtlinie hinweisen, und 404-Fehlercodes können beispielsweise auf ein fehlerhaftes Verhalten der Client-Anwendung hinweisen. Die vorübergehende Aktivierung der Protokollierung des S3-Serverzugriffs hilft Ihnen, die Ursache des Problems anhand der Felder HTTP-Status und Fehlercode zu ermitteln. Weitere Informationen zum Fehlercode finden Sie unter Fehlerantworten.

Absicht: Dieser Alarm wird verwendet, um eine Baseline für typische 4XX-Fehlerraten zu erstellen, sodass Sie alle Auffälligkeiten untersuchen können, die auf ein Einrichtungsproblem hinweisen könnten.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 0,05

Begründung des Schwellenwerts: Der empfohlene Schwellenwert besteht darin, zu erkennen, ob bei mehr als 5 % der gesamten Anfragen 4XX-Fehler auftreten. Bei häufig auftretenden 4XX-Fehlern sollte ein Alarm ausgelöst werden. Die Einstellung eines sehr niedrigen Schwellenwerts kann jedoch dazu führen, dass der Alarm zu empfindlich ist. Sie können den Schwellenwert auch an die Auslastung der Anfragen anpassen und dabei eine akzeptable Anzahl von 4XX-Fehlern berücksichtigen. Sie können auch historische Daten analysieren, um die akzeptable Fehlerrate für den Anwendungs-Workload zu ermitteln, und den Schwellenwert dann entsprechend anpassen.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: GREATER_THAN_THRESHOLD

5xxErrors

Abmessungen:BucketName, FilterId

Alarmbeschreibung: Dieser Alarm hilft Ihnen, eine große Anzahl von serverseitigen Fehlern zu erkennen. Diese Fehler deuten darauf hin, dass ein Client eine Anfrage gestellt hat, die der Server nicht abschließen konnte. So können Sie das Problem, mit dem Ihre Anwendung aufgrund von S3 konfrontiert ist, besser zuordnen. Weitere Informationen, die Ihnen helfen, Fehler effizient zu behandeln oder zu reduzieren, finden Sie unter Optimieren von Leistungsentwurfsmustern. Die Fehler können auch durch ein Problem mit S3 verursacht werden. Überprüfen Sie den Status von Amazon S3 in Ihrer Region im AWS Service Health Dashboard.

Absicht: Dieser Alarm kann dabei helfen, festzustellen, ob bei der Anwendung Probleme aufgrund von 5XX-Fehlern auftreten.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 0,05

Begründung des Schwellenwerts: Wir empfehlen, den Schwellenwert so festzulegen, dass erkannt wird, ob mehr als 5 % aller Anfragen 5XXError aufweisen. Sie können den Schwellenwert jedoch an den Datenverkehr der Anfragen sowie an die akzeptablen Fehlerraten anpassen. Sie können auch historische Daten analysieren, um festzustellen, welche Fehlerrate für den Anwendungs-Workload akzeptabel ist, und den Schwellenwert entsprechend anpassen.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: GREATER_THAN_THRESHOLD

OperationsFailedReplication

Abmessungen:SourceBucket, DestinationBucket, RuleId

Alarmbeschreibung: Dieser Alarm hilft beim Verständnis eines Replikationsfehlers. Diese Metrik verfolgt den Status neuer Objekte, die mit S3 CRR oder S3 SRR repliziert wurden, sowie vorhandener Objekte, die mit der S3-Batchreplikation repliziert wurden. Weitere Informationen finden Sie unter Problembehandlung bei der Replikation.

Absicht: Dieser Alarm wird verwendet, um zu erkennen, ob ein Replikationsvorgang fehlgeschlagen ist.

Statistik: Maximum

Empfohlener Schwellenwert: 0,0

Begründung des Schwellenwerts: Diese Metrik gibt bei erfolgreichen Vorgängen den Wert 0 aus, und nichts, wenn für die Minute keine Replikationsvorgänge ausgeführt wurden. Wenn die Metrik einen Wert größer als 0 ausgibt, ist der Replikationsvorgang nicht erfolgreich.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

S3ObjectLambda

4xxErrors

Abmessungen:AccessPointName, DataSource ARN

Beschreibung des Alarms: Dieser Alarm hilft uns, die Gesamtzahl der 4XX-Fehlerstatuscodes zu melden, die als Antwort auf Kundenanfragen ausgegeben wurden. Die vorübergehende Aktivierung der Protokollierung des S3-Serverzugriffs hilft Ihnen, die Ursache des Problems anhand der Felder HTTP-Status und Fehlercode zu ermitteln.

Absicht: Dieser Alarm wird verwendet, um eine Baseline für typische 4XX-Fehlerraten zu erstellen, sodass Sie alle Auffälligkeiten untersuchen können, die auf ein Einrichtungsproblem hinweisen könnten.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 0,05

Begründung des Schwellenwerts: Wir empfehlen, den Schwellenwert so festzulegen, dass erkannt wird, ob mehr als 5 % aller Anfragen 4XXError aufweisen. Bei häufig auftretenden 4XX-Fehlern sollte ein Alarm ausgelöst werden. Die Einstellung eines sehr niedrigen Schwellenwerts kann jedoch dazu führen, dass der Alarm zu empfindlich ist. Sie können den Schwellenwert auch an die Auslastung der Anfragen anpassen und dabei eine akzeptable Anzahl von 4XX-Fehlern berücksichtigen. Sie können auch historische Daten analysieren, um die akzeptable Fehlerrate für den Anwendungs-Workload zu ermitteln, und den Schwellenwert dann entsprechend anpassen.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: GREATER_THAN_THRESHOLD

5xxErrors

Abmessungen:AccessPointName, DataSource ARN

Alarmbeschreibung: Dieser Alarm hilft dabei, eine große Anzahl von serverseitigen Fehlern zu erkennen. Diese Fehler deuten darauf hin, dass ein Client eine Anfrage gestellt hat, die der Server nicht abschließen konnte. Diese Fehler können durch ein Problem mit S3 verursacht werden. Überprüfen Sie den Status von Amazon S3 in Ihrer Region im AWS Service Health Dashboard. So können Sie das Problem, mit dem Ihre Anwendung aufgrund von S3 konfrontiert ist, besser zuordnen. Informationen, die Ihnen helfen, diese Fehler effizient zu behandeln oder zu reduzieren, finden Sie unter Optimieren von Leistungsentwurfsmustern.

Absicht: Dieser Alarm kann dabei helfen, festzustellen, ob bei der Anwendung Probleme aufgrund von 5XX-Fehlern auftreten.

Statistik: Durchschnitt

Empfohlener Schwellenwert: 0,05

Begründung des Schwellenwerts: Wir empfehlen, den Schwellenwert so festzulegen, dass erkannt wird, ob mehr als 5 % aller Anfragen 5XXError aufweisen. Sie können den Schwellenwert jedoch an den Datenverkehr der Anfragen sowie an die akzeptablen Fehlerraten anpassen. Sie können auch historische Daten analysieren, um festzustellen, welche Fehlerrate für den Anwendungs-Workload akzeptabel ist, und den Schwellenwert entsprechend anpassen.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: GREATER_THAN_THRESHOLD

LambdaResponse4x

Abmessungen:AccessPointName, DataSource ARN

Alarmbeschreibung: Dieser Alarm hilft Ihnen, Fehler (500s) bei Aufrufen von S3 Object Lambda zu erkennen und zu diagnostizieren. Diese Fehler können durch Fehler oder Fehlkonfigurationen in der Lambda-Funktion verursacht werden, die für die Beantwortung Ihrer Anfragen verantwortlich ist. Wenn CloudWatch Sie die Log-Streams der Lambda-Funktion untersuchen, die dem Object Lambda Access Point zugeordnet ist, können Sie anhand der Antwort von S3 Object Lambda den Ursprung des Problems ermitteln.

Absicht: Dieser Alarm wird verwendet, um 4xx-Client-Fehler bei Aufrufen zu erkennen. WriteGetObjectResponse

Statistik: Durchschnitt

Empfohlener Schwellenwert: 0,05

Begründung des Schwellenwerts: Wir empfehlen, den Schwellenwert so festzulegen, dass erkannt wird, ob mehr als 5 % aller Anfragen 4XXError aufweisen. Bei häufig auftretenden 4XX-Fehlern sollte ein Alarm ausgelöst werden. Die Einstellung eines sehr niedrigen Schwellenwerts kann jedoch dazu führen, dass der Alarm zu empfindlich ist. Sie können den Schwellenwert auch an die Auslastung der Anfragen anpassen und dabei eine akzeptable Anzahl von 4XX-Fehlern berücksichtigen. Sie können auch historische Daten analysieren, um die akzeptable Fehlerrate für den Anwendungs-Workload zu ermitteln, und den Schwellenwert dann entsprechend anpassen.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: GREATER_THAN_THRESHOLD

Amazon SNS

NumberOfMessagesPublished

Abmessungen: TopicName

Alarmbeschreibung: Dieser Alarm kann erkennen, wenn die Anzahl der veröffentlichten SNS-Nachrichten zu gering ist. Überprüfen Sie zur Fehlerbehebung, warum die Publisher weniger Datenverkehr senden.

Absicht: Dieser Alarm hilft Ihnen dabei, proaktiv zu überwachen und signifikante Rückgänge bei der Veröffentlichung von Benachrichtigungen zu erkennen. Auf diese Weise können Sie potenzielle Probleme mit Ihren Anwendungen oder Geschäftsprozessen identifizieren, sodass Sie geeignete Maßnahmen ergreifen können, um den erwarteten Benachrichtigungsfluss aufrechtzuerhalten. Sie sollten diesen Alarm erstellen, wenn Sie davon ausgehen, dass Ihr System ein Minimum an Datenverkehr bedient.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Die Anzahl der veröffentlichten Nachrichten sollte der erwarteten Anzahl veröffentlichter Nachrichten für Ihre Anwendung entsprechen. Sie können auch die historischen Daten, Trends und den Datenverkehr analysieren, um den richtigen Schwellenwert zu finden.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: LESS_THAN_THRESHOLD

NumberOfNotificationsDelivered

Abmessungen: TopicName

Alarmbeschreibung: Dieser Alarm kann erkennen, wenn die Anzahl der übermittelten SNS-Nachrichten zu gering ist. Dies kann auf die unbeabsichtigte Abmeldung eines Endpunkts oder auf ein SNS-Ereignis zurückzuführen sein, das zu Verzögerungen bei Nachrichten führt.

Absicht: Dieser Alarm hilft Ihnen dabei, einen Rückgang der zugestellten Nachrichtenmenge zu erkennen. Sie sollten diesen Alarm erstellen, wenn Sie davon ausgehen, dass Ihr System ein Minimum an Datenverkehr bedient.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Die Anzahl der zugestellten Nachrichten sollte der erwarteten Anzahl der produzierten Nachrichten und der Anzahl der Verbraucher entsprechen. Sie können auch die historischen Daten, Trends und den Datenverkehr analysieren, um den richtigen Schwellenwert zu finden.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: LESS_THAN_THRESHOLD

NumberOfNotificationsFailed

Abmessungen: TopicName

Alarmbeschreibung: Dieser Alarm kann erkennen, wenn die Anzahl fehlgeschlagener SNS-Nachrichten zu hoch ist. Um Fehler bei Benachrichtigungen zu beheben, aktivieren Sie die Protokollierung in CloudWatch Logs. Wenn Sie die Protokolle überprüfen, können Sie herausfinden, welche Subscriber ausfallen und welche Statuscodes sie zurückgeben.

Absicht: Dieser Alarm hilft Ihnen dabei, proaktiv Probleme bei der Zustellung von Benachrichtigungen zu finden und geeignete Maßnahmen zu ihrer Behebung zu ergreifen.

Statistik: Summe

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Der empfohlene Schwellenwert für diesen Alarm hängt stark von den Auswirkungen fehlgeschlagener Benachrichtigungen ab. Prüfen Sie die für Ihre Endbenutzer bereitgestellten SLAs, die Fehlertoleranz und die Wichtigkeit der Benachrichtigungen, analysieren Sie historische Daten und wählen Sie dann einen entsprechenden Schwellenwert aus. Die Anzahl der fehlgeschlagenen Benachrichtigungen sollte für Themen, die nur SQS-, Lambda- oder Firehose-Abonnements haben, 0 sein.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

NumberOfNotificationsFilteredOut-InvalidAttributes

Abmessungen: TopicName

Alarmbeschreibung: Dieser Alarm hilft dabei, potenzielle Probleme mit dem Publisher oder den Subscribern zu überwachen und zu lösen. Prüfen Sie, ob ein Publisher Nachrichten mit ungültigen Attributen veröffentlicht oder ob ein unangemessener Filter auf einen Subscriber angewendet wird. Sie können auch CloudWatch Protokolle analysieren, um die Ursache des Problems zu finden.

Absicht: Dieser Alarm wird verwendet, um festzustellen, ob die veröffentlichten Nachrichten nicht gültig sind oder ob unangemessene Filter auf einen Subscriber angewendet wurden.

Statistik: Summe

Empfohlener Schwellenwert: 0,0

Begründung des Schwellenwerts: Ungültige Attribute sind fast immer ein Fehler des Publishers. Wir empfehlen, den Schwellenwert auf 0 zu setzen, da in einem fehlerfreien System keine ungültigen Attribute zu erwarten sind.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

NumberOfNotificationsFilteredOut-InvalidMessageBody

Abmessungen: TopicName

Alarmbeschreibung: Dieser Alarm hilft dabei, potenzielle Probleme mit dem Publisher oder den Subscribern zu überwachen und zu lösen. Prüfen Sie, ob ein Publisher Nachrichten mit ungültigen Nachrichtentexten veröffentlicht oder ob ein unangemessener Filter auf einen Subscriber angewendet wird. Sie können auch CloudWatch Protokolle analysieren, um die Ursache des Problems zu finden.

Absicht: Dieser Alarm wird verwendet, um festzustellen, ob die veröffentlichten Nachrichten nicht gültig sind oder ob unangemessene Filter auf einen Subscriber angewendet wurden.

Statistik: Summe

Empfohlener Schwellenwert: 0,0

Begründung des Schwellenwerts: Ungültige Nachrichtentexte sind fast immer ein Fehler des Publisher. Wir empfehlen, den Schwellenwert auf 0 zu setzen, da in einem fehlerfreien System keine ungültigen Nachrichtentexte zu erwarten sind.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

NumberOfNotificationsRedrivenToDlq

Abmessungen: TopicName

Alarmbeschreibung: Dieser Alarm hilft dabei, die Anzahl der Nachrichten zu überwachen, die in eine Warteschlange für unzustellbare Nachrichten verschoben wurden.

Absicht: Dieser Alarm wird verwendet, um Nachrichten zu erkennen, die in eine Warteschlange für unzustellbare Nachrichten verschoben wurden. Wir empfehlen, diesen Alarm auszulösen, wenn SNS mit SQS, Lambda oder Firehose gekoppelt ist.

Statistik: Summe

Empfohlener Schwellenwert: 0,0

Begründung des Schwellenwerts: In einem fehlerfreien System, egal welchen Subscribertyps, sollten Nachrichten nicht in die Warteschlange für unzustellbare Nachrichten verschoben werden. Wir empfehlen, dass Sie benachrichtigt werden, wenn Nachrichten in der Warteschlange landen, damit Sie die Ursache identifizieren und beheben und die Nachrichten in der Warteschlange für unzustellbare Nachrichten möglicherweise erneut versenden können, um Datenverlust zu verhindern.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

NumberOfNotificationsFailedToRedriveToDlq

Abmessungen: TopicName

Alarmbeschreibung: Dieser Alarm hilft bei der Überwachung von Nachrichten, die nicht in eine Warteschlange für unzustellbare Nachrichten verschoben werden konnten. Prüfen Sie, ob Ihre Warteschlange für unzustellbare Nachrichten existiert und ob sie richtig konfiguriert ist. Stellen Sie außerdem sicher, dass SNS über Berechtigungen für den Zugriff auf die Warteschlange für unzustellbare Nachrichten verfügt. Weitere Informationen finden Sie in der Dokumentation zur Warteschlange für unzustellbare Nachrichten.

Absicht: Dieser Alarm wird verwendet, um Nachrichten zu erkennen, die nicht in eine Warteschlange für unzustellbare Nachrichten verschoben werden konnten.

Statistik: Summe

Empfohlener Schwellenwert: 0,0

Begründung des Schwellenwerts: Es ist fast immer ein Fehler, wenn Nachrichten nicht in die Warteschlange für unzustellbare Nachrichten verschoben werden können. Die Empfehlung für den Schwellenwert ist 0, was bedeutet, dass alle Nachrichten, deren Verarbeitung fehlschlägt, in der Lage sein müssen, die Warteschlange für unzustellbare Nachrichten zu erreichen, wenn die Warteschlange konfiguriert wurde.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

SMS MonthToDateSpent USD

Abmessungen: TopicName

Alarmbeschreibung: Mithilfe des Alarms können Sie überwachen, ob Ihr Konto über ein ausreichendes Kontingent verfügt, damit SNS Nachrichten zustellen kann. Wenn Sie Ihr Kontingent erreichen, kann SNS keine SMS-Nachrichten zustellen. Informationen zum Einrichten Ihres monatlichen SMS-Ausgabenkontingents oder zur Beantragung einer Erhöhung des Ausgabenkontingents mit AWS finden Sie unter Einstellungen für SMS-Nachrichten festlegen.

Absicht: Dieser Alarm wird verwendet, um festzustellen, ob Ihr Konto über ein ausreichendes Kontingent verfügt, damit Ihre SMS-Nachrichten erfolgreich zugestellt werden können.

Statistik: Maximum

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Legen Sie den Schwellenwert entsprechend dem Kontingent (Ausgabenlimit des Kontos) für das Konto fest. Wählen Sie einen Schwellenwert, der Sie früh genug darüber informiert, dass Sie Ihr Kontingentlimit erreicht haben, sodass Sie Zeit haben, eine Erhöhung zu beantragen.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

SMS SuccessRate

Abmessungen: TopicName

Alarmbeschreibung: Dieser Alarm hilft dabei, die Rate fehlgeschlagener SMS-Nachrichtenzustellungen zu überwachen. Sie können Cloudwatch Logs einrichten, um die Art des Fehlers zu verstehen und auf dieser Grundlage Maßnahmen zu ergreifen.

Absicht: Dieser Alarm wird verwendet, um fehlgeschlagene SMS-Nachrichtenzustellungen zu erkennen.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Stellen Sie den Schwellenwert für den Alarm entsprechend Ihrer Toleranz für fehlgeschlagene SMS-Nachrichtenzustellungen ein.

Zeitraum: 60

Datenpunkte bis Alarm: 5

Auswertungszeiträume: 5

Vergleichsoperator: GREATER_THAN_THRESHOLD

Amazon SQS

ApproximateAgeOfOldestMessage

Abmessungen: QueueName

Alarmbeschreibung: Dieser Alarm überwacht das Alter der ältesten Nachricht in der Warteschlange. Sie können diesen Alarm verwenden, um zu überwachen, ob Ihre Verbraucher SQS-Nachrichten mit der gewünschten Geschwindigkeit verarbeiten. Erwägen Sie, die Anzahl der Verbraucher oder den Durchsatz der Verbraucher zu erhöhen, um das Alter der Nachrichten zu verringern. Diese Metrik kann in Kombination mit `ApproximateNumberOfMessagesVisible` verwendet werden, um zu bestimmen, wie groß der Warteschlangenrückstand ist und wie schnell Nachrichten verarbeitet werden. Um zu verhindern, dass Nachrichten vor der Verarbeitung gelöscht werden, sollten Sie in Erwägung ziehen, die Warteschlange für unzustellbare

Nachrichten so zu konfigurieren, dass potenzielle Poison-Pill-Nachrichten außer Acht gelassen werden.

Absicht: Dieser Alarm wird verwendet, um zu erkennen, ob das Alter der ältesten Nachricht in der QueueName Warteschlange zu hoch ist. Ein hohes Alter kann ein Hinweis darauf sein, dass Nachrichten nicht schnell genug verarbeitet werden oder dass einige Poison-Pill-Nachrichten in der Warteschlange stecken bleiben und nicht verarbeitet werden können.

Statistik: Maximum

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Der empfohlene Schwellenwert für diesen Alarm hängt stark von der erwarteten Nachrichtenverarbeitungszeit ab. Sie können historische Daten verwenden, um die durchschnittliche Nachrichtenverarbeitungszeit zu berechnen, und dann den Schwellenwert auf einen Wert festlegen, der 50 % über der von den Warteschlangenverbrauchern erwarteten maximalen Verarbeitungszeit für SQS-Nachrichten liegt.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

ApproximateNumberOfMessagesNotVisible

Abmessungen: QueueName

Alarmbeschreibung: Dieser Alarm hilft bei der Erkennung einer hohen Anzahl von Nachrichten während der Übertragung in Bezug auf `QueueName`. Zur Fehlerbehebung prüfen Sie Nachrichtenrückstand nimmt ab.

Absicht: Dieser Alarm wird verwendet, um eine hohe Anzahl von Nachrichten während der Übertragung in der Warteschlange zu erkennen. Wenn Verbraucher Nachrichten nicht innerhalb des Sichtbarkeitszeitlimits löschen, werden die Nachrichten beim Abrufen der Warteschlange wieder in der Warteschlange angezeigt. Bei FIFO-Warteschlangen können maximal 20 000 Nachrichten während der Übertragung gespeichert werden. Wenn Sie dieses Kontingent erreichen, gibt SQS keine Fehlermeldungen zurück. Eine FIFO-Warteschlange durchsucht die ersten 20 000 Nachrichten, um die verfügbaren Nachrichtengruppen zu ermitteln. Das heißt, wenn Sie in einer einzelnen Nachrichtengruppe einen Rückstand an Nachrichten haben, können

Sie Nachrichten aus anderen Nachrichtengruppen, die zu einem späteren Zeitpunkt an die Warteschlange gesendet wurden, erst verarbeiten, wenn Sie die Nachrichten aus dem Rückstand erfolgreich verarbeitet haben.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Der empfohlene Schwellenwert für diesen Alarm hängt stark von der zu erwartenden Anzahl an Nachrichten während der Übertragung ab. Sie können historische Daten verwenden, um die maximal zu erwartende Anzahl von Nachrichten während der Übertragung zu berechnen und den Schwellenwert auf 50 % über diesem Wert festzulegen. Wenn Verbraucher der Warteschlange Nachrichten verarbeiten, aber nicht aus der Warteschlange löschen, steigt diese Zahl plötzlich an.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

ApproximateNumberOfMessagesVisible

Abmessungen: QueueName

Alarmbeschreibung: Dieser Alarm überwacht, ob der Rückstand in der Nachrichtenwarteschlange größer als erwartet ist, was darauf hindeutet, dass die Verbraucher zu langsam sind oder nicht genügend Verbraucher vorhanden sind. Ziehen Sie in Erwägung, die Anzahl der Verbraucher zu erhöhen oder die Verbraucher zu beschleunigen, wenn dieser Alarm in den Zustand ALARM übergeht.

Absicht: Dieser Alarm wird verwendet, um zu erkennen, ob die Anzahl der Nachrichten in der aktiven Warteschlange zu hoch ist und die Verbraucher die Nachrichten nur langsam verarbeiten, oder ob nicht genügend Verbraucher vorhanden sind, um sie zu verarbeiten.

Statistik: Durchschnitt

Empfohlener Schwellenwert: Hängt von Ihrer Situation ab

Begründung des Schwellenwerts: Eine unerwartet hohe Anzahl sichtbarer Nachrichten weist darauf hin, dass Nachrichten von einem Verbraucher nicht mit der erwarteten Geschwindigkeit

verarbeitet werden. Bei der Festlegung dieses Schwellenwerts sollten Sie historische Daten berücksichtigen.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

NumberOfMessagesSent

Abmessungen: QueueName

Alarmbeschreibung: Dieser Alarm hilft zu erkennen, ob von einem Produzenten keine Nachrichten in Bezug auf `QueueName` gesendet wurden. Überprüfen Sie zur Fehlerbehebung den Grund, warum der Produzent keine Nachrichten sendet.

Absicht: Dieser Alarm wird verwendet, um zu erkennen, wenn ein Produzent keine Nachrichten mehr sendet.

Statistik: Summe

Empfohlener Schwellenwert: 0,0

Begründung des Schwellenwerts: Wenn die Anzahl der gesendeten Nachrichten 0 ist, sendet der Produzent keine Nachrichten. Wenn diese Warteschlange einen niedrigen TPS-Wert hat, erhöhen Sie die Anzahl der EvaluationPeriods entsprechend.

Zeitraum: 60

Datenpunkte bis Alarm: 15

Auswertungszeiträume: 15

Vergleichsoperator: LESS_THAN_OR_EQUAL_TO_THRESHOLD

AWS VPN

TunnelState

Abmessungen: VpnId

Alarmbeschreibung: Dieser Alarm hilft Ihnen zu verstehen, ob der Status eines oder mehrerer Tunnel INAKTIV ist. Informationen zur Fehlerbehebung finden Sie unter Fehlerbehebung bei VPN-Tunneln.

Absicht: Dieser Alarm wird verwendet, um festzustellen, ob sich mindestens ein Tunnel für dieses VPN im Status INAKTIV befindet, sodass Sie eine Fehlerbehebung für das betroffene VPN durchführen können. Dieser Alarm befindet sich bei Netzwerken, in denen nur ein einziger Tunnel konfiguriert ist, immer im ALARM-Status.

Statistik: Minimum

Empfohlener Schwellenwert: 1,0

Begründung des Schwellenwerts: Ein Wert unter 1 gibt an, dass sich mindestens ein Tunnel im Status INAKTIV befindet.

Zeitraum: 300

Datenpunkte bis Alarm: 3

Auswertungszeiträume: 3

Vergleichsoperator: LESS_THAN_THRESHOLD

TunnelState

Abmessungen: TunnelIpAddress

Alarmbeschreibung: Dieser Alarm hilft Ihnen zu verstehen, ob der Status dieses Tunnels INAKTIV ist. Informationen zur Fehlerbehebung finden Sie unter Fehlerbehebung bei VPN-Tunneln.

Absicht: Dieser Alarm wird verwendet, um zu erkennen, ob sich der Tunnel im Zustand INAKTIV befindet, so dass Sie das betroffene VPN auf Fehler untersuchen können. Dieser Alarm befindet sich bei Netzwerken, in denen nur ein einziger Tunnel konfiguriert ist, immer im ALARM-Status.

Statistik: Minimum

Empfohlener Schwellenwert: 1,0

Begründung des Schwellenwerts: Ein Wert unter 1 gibt an, dass sich der Tunnel im Status INAKTIV befindet.

Zeitraum: 300

Datenpunkte bis Alarm: 3

Auswertungszeiträume: 3

Vergleichsoperator: LESS_THAN_THRESHOLD

Alarmieren bei Metriken

In den folgenden Abschnitten wird erläutert, wie CloudWatch Alarme für Messwerte erstellt werden.

Erstellen Sie einen CloudWatch Alarm auf der Grundlage eines statischen Schwellenwerts

Sie wählen eine CloudWatch Metrik für den Alarm, der überwacht werden soll, und den Schwellenwert für diese Metrik. Der Alarm wechselt in den Status `ALARM`, wenn die Metrik für eine bestimmte Anzahl von Auswertungszeiträumen den Schwellenwert überschreitet.

Wenn Sie einen Alarm in einem Konto erstellen, das als Überwachungskonto für CloudWatch kontenübergreifende Beobachtbarkeit eingerichtet wurde, können Sie den Alarm so einrichten, dass eine Metrik in einem Quellkonto überwacht wird, das mit diesem Überwachungskonto verknüpft ist. Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.

So erstellen Sie einen Alarm basierend auf einer einzelnen Metrik

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Alarms (Alarme) und All alarms (Alle Alarme) aus.
3. Wählen Sie Create alarm (Alarm erstellen) aus.
4. Wählen Sie Select Metric (Metrik auswählen) aus.
5. Führen Sie eine der folgenden Aktionen aus:
   - Wählen Sie den Service-Namespace, der die gewünschte Metrik enthält. Fahren Sie mit der Auswahl der sichtbar werdenden Optionen fort, um die Auswahl einzugrenzen. Wenn eine Liste von Metriken angezeigt wird, aktivieren Sie das Kontrollkästchen neben der gewünschten Metrik.
   - Geben Sie in das Suchfeld den Namen einer Metrik, Konto-ID, Kontobezeichnung, Dimension oder Ressourcen-ID ein. Wählen Sie dann eines der Ergebnisse aus und fahren

Sie fort, bis eine Liste von Metriken erscheint. Aktivieren Sie das Kontrollkästchen neben der gewünschten Metrik.

6. Wählen Sie die Registerkarte Graphed metrics (Grafisch dargestellte Metriken) aus.

   a. Wählen Sie unter Statistic (Statistik) eine der Statistiken oder vordefinierten Perzentile aus, oder geben Sie ein benutzerdefiniertes Perzentil an (z. B. **`p95.45`**).

   b. Wählen Sie unter Period (Zeitraum) den Auswertungszeitraum für den Alarm aus. Beim Auswerten des Alarms wird jeder Zeitraum in einem Datenpunkt zusammengefasst.

      Sie können auch wählen, ob die Legende der y-Achse während der Erstellung des Alarms links oder rechts angezeigt wird. Diese Einstellung wird nur verwendet, wenn Sie den Alarm erstellen.

   c. Wählen Sie Select Metric (Metrik auswählen) aus.

      Die Seite Specify metric and conditions (Metrik und Bedingungen festlegen) wird angezeigt, die ein Diagramm und andere Informationen über die von Ihnen ausgewählte Metrik und Statistik anzeigt.

7. Geben Sie unter Conditions (Bedingungen) Folgendes an:

   a. Geben Sie für Wann immer die ***`Metrik`*** ist an, ob die Metrik größer, kleiner oder gleich dem Schwellenwert sein muss. Geben Sie unter than... (dann ...) den Schwellenwert an.

   b. Wählen Sie Additional configuration (Zusätzliche Konfiguration). Geben Sie unter Datapoints to alarm (Datenpunkte für Alarm) an, wie viele Auswertungszeiträume (Datenpunkte) im Status `ALARM` sein müssen, damit der Alarm ausgelöst wird. Wenn die beiden Werte hier übereinstimmen, erstellen Sie einen Alarm, der in den Status `ALARM` wechselt, wenn entsprechend viele aufeinanderfolgende Zeiträume überschritten werden.

      Um einen M aus N Alarm zu erstellen, geben Sie eine niedrigere Zahl für den ersten Wert als für den zweiten Wert an. Weitere Informationen finden Sie unter Auswerten eines Alarms.

   c. Wählen Sie für Missing data treatment (Behandlung von fehlenden Daten) aus, wie sich der Alarm verhalten soll, wenn einige Datenpunkte fehlen. Weitere Informationen finden Sie unter Konfiguration, wie Alarme fehlende Daten behandeln CloudWatch .

   d. Wenn der Alarm ein Perzentil als überwachte Statistik verwendet, erscheint ein Feld Percentiles with low samples (Perzentile mit geringen Stichproben). Verwenden Sie diese Option, um zu entscheiden, ob Sie Fälle mit niedrigem Stichprobenumfang bewerten oder ignorieren möchten. Wenn Sie ignore (maintain alarm state) (Ignorieren (Alarmstatus

beibehalten)) wählen, wird der aktuelle Alarmstatus immer beibehalten, wenn die Stichprobengröße zu gering ist. Weitere Informationen finden Sie unter Auf Perzentilen basierende CloudWatch Alarme und Stichproben mit niedrigen Datenmengen.

8. Wählen Sie Weiter.
9. Wählen Sie unter Notification (Benachrichtigung) ein SNS-Thema aus, das benachrichtigt werden soll, wenn sich der Alarm im Status `ALARM`, `OK` oder `INSUFFICIENT_DATA` befindet.

   Um zu erreichen, dass der Alarm mehrere Benachrichtigungen für den gleichen Alarmstatus oder für verschiedene Statuswerte sendet, wählen Sie Benachrichtigung hinzufügen.

   Bei der CloudWatch kontoübergreifenden Beobachtbarkeit können Sie festlegen, dass Benachrichtigungen an mehrere AWS Konten gesendet werden. Zum Beispiel sowohl für das Überwachungskonto als auch für das Quellkonto.

   Damit der Alarm keine Benachrichtigungen sendet, wählen Sie Remove (Entfernen).
10. Um den Alarm Auto-Scaling-, EC2-, Lambda- oder Systems-Manager-Aktionen ausführen zu lassen, wählen Sie die entsprechende Schaltfläche und wählen Sie den Alarmstatus und die auszuführende Aktion. Alarme können Aktionen des Systems Manager nur ausführen, wenn sie in den ALARM-Zustand wechseln. Weitere Informationen zu Systems Manager Manager-Aktionen finden Sie unter Konfiguration für CloudWatch die Erstellung OpsItems aus Alarmen und Incident-Erstellung.

    > **Note**
    >
    > Um einen Alarm zu erstellen, der eine SSM-Incident-Manager-Aktion ausführt, müssen Sie über bestimmte Berechtigungen verfügen. Weitere Informationen finden Sie unter Beispiele für identitätsbasierte Richtlinien für AWS Systems Manager Incident Manager.
11. Wenn Sie fertig sind, wählen Sie Weiter.
12. Geben Sie einen Namen und eine Beschreibung für den Alarm ein. Der Name darf nur UTF-8-Zeichen und keine ASCII-Kontrolleingabezeichen enthalten. Die Beschreibung kann Markdown-Formatierungen enthalten, die nur auf der Registerkarte Alarmdetails in der Konsole angezeigt werden. CloudWatch Der Markdown kann nützlich sein, um Links zu Runbooks oder anderen internen Ressourcen hinzuzufügen. Wählen Sie anschließend Weiter.
13. Bestätigen Sie unter Preview and create (Vorschau und erstellen), dass die Informationen und Bedingungen den Anforderungen entsprechen, und wählen Sie dann Create alarm (Alarm erstellen).

Sie können auch Alarme zu einem Dashboard hinzufügen. Weitere Informationen finden Sie unter Fügen Sie ein Alarm-Widget zu einem CloudWatch Dashboard hinzu oder entfernen Sie es .

Erstellen Sie einen CloudWatch Alarm auf der Grundlage eines metrischen mathematischen Ausdrucks

Um einen Alarm auf der Grundlage eines metrischen mathematischen Ausdrucks zu erstellen, wählen Sie eine oder mehrere CloudWatch Metriken aus, die in dem Ausdruck verwendet werden sollen. Geben Sie dann den Ausdruck, den Schwellenwert und die Auswertungszeiträume an.

Sie können keinen Alarm basierend auf dem SEARCH-Ausdruck erstellen. Dies liegt daran, dass Suchausdrücke mehrere Zeitreihen zurückgeben und ein Alarm, der auf einem mathematischen Ausdruck basiert, nur eine Zeitreihe beobachten kann.

So erstellen Sie einen Alarms basierend auf einem Metrikberechnungs-Ausdruck:

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Alarme und dann Alle Alarme aus.
3. Wählen Sie Alarm erstellen aus.
4. Wählen Sie Select Metric (Metrik auswählen) aus und führen Sie dann eine der folgenden Aktionen durch:
   - Wählen Sie im Dropdown–Menü AWS -Namespaces oder Custom namespaces (Benutzerdefinierte Namespaces) einen Namespace aus. Nachdem Sie einen Namespace ausgewählt haben, wählen Sie weitere Optionen aus, bis eine Liste von Metriken angezeigt wird, in der Sie das Kontrollkästchen neben der gewünschten Metrik aktivieren.
   - Verwenden Sie das Suchfeld, um eine Metrik, Konto-ID, Dimension oder Ressourcen-ID zu suchen. Nachdem Sie die Metrik, Dimension oder Ressourcen-ID eingegeben haben, wählen Sie weitere Optionen aus, bis eine Liste von Metriken angezeigt wird, in der Sie das Kontrollkästchen neben der gewünschten Metrik aktivieren.
5. (Optional) Wenn Sie einem Metrikberechnungs-Ausdruck eine weitere Metrik hinzufügen möchten, können Sie das Suchfeld verwenden, um eine bestimmte Metrik zu suchen. Sie können einem Metrikberechnungs-Ausdruck bis zu 10 Metriken hinzufügen.
6. Wählen Sie die Registerkarte Graphed metrics (Grafisch dargestellte Metriken) aus. Führen Sie für alle Metriken, die Sie zuvor hinzugefügt haben, die folgenden Aktionen durch:

a. Wählen Sie unter der Spalte Statistic (Statistik) das Dropdown-Menü aus. Wählen Sie im Dropdown-Menü eine der vordefinierten Statistiken oder ein Perzentil aus. Verwenden Sie das Suchfeld im Dropdown-Menü, um ein benutzerdefiniertes Perzentil anzugeben.

b. Wählen Sie unter der Spalte Zeitraum das Dropdown-Menü aus. Wählen Sie im Dropdown-Menü einen der vordefinierten Auswertungszeiträume aus.

   Beim Erstellen des Alarms können Sie auch bestimmen, ob die Legende der y-Achse links oder rechts vom Diagramm angezeigt werden soll.

> **Note**
>
> Bei der CloudWatch Auswertung von Alarmen werden Zeiträume zu einzelnen Datenpunkten zusammengefasst.

7. Wählen Sie das Dropdown-Menü Add math (Berechnung hinzufügen) und dann in der Liste der vordefinierten Metrikberechnungs-Ausdrücke die Option Start with an empty expression (Mit leerem Ausdruck beginnen) aus.

   Nachdem Sie Start with an empty expression (Mit einem leeren Ausdruck beginnen) ausgewählt haben, erscheint ein Feld für mathematische Ausdrücke, in dem Sie mathematische Ausdrücke anwenden oder bearbeiten können.

8. Geben Sie im Feld für mathematische Ausdrücke Ihren mathematischen Ausdruck ein und wählen Sie dann Apply (Anwenden) aus.

   Nachdem Sie Apply (Anwenden) ausgewählt haben, erscheint die Spalte ID neben der Spalte Label (Bezeichnung).

   Um eine Metrik oder das Ergebnis eines anderen Metrikberechnungs-Ausdrucks in der Formel des aktuellen mathematischen Ausdrucks zu nutzen, verwenden Sie den in der Spalte ID angezeigten Wert. Um den Wert von ID zu ändern, wählen Sie das pen-and-paper Symbol neben dem aktuellen Wert aus. Der neue Wert muss mit einem Kleinbuchstaben beginnen und kann Ziffern, Buchstaben und Unterstriche enthalten. Durch Ändern des Wertes von ID auf einen aussagekräftigeren Namen können Sie das Alarmdiagramm besser verständlich machen.

   Informationen zu den Funktionen, die für Metrikberechnungen verfügbar sind, finden Sie unter Syntax und Funktionen von Metrikberechnungen.

9. (Optional), fügen Sie weitere mathematische Ausdrücke hinzu, indem Sie sowohl Metriken als auch die Ergebnisse anderer mathematischer Ausdrücke in den Formeln der neuen mathematischen Ausdrücke verwenden.
10. Wenn Sie den Ausdruck für den Alarm verwenden möchten, deaktivieren Sie die Kontrollkästchen links neben jedem anderen Ausdruck und jeder Metrik auf der Seite. Nur das Kontrollkästchen neben dem Ausdruck, der für den Alarm verwendet werden soll, sollte aktiviert sein. Der Ausdruck, den Sie für den Alarm wählen, muss eine einzige Zeitreihe ergeben und nur eine Zeile im Diagramm anzeigen. Wählen Sie dann Select Metric (Metrik auswählen) aus.

    Die Seite Specify metric and conditions (Metrik und Bedingungen festlegen) wird angezeigt, die ein Diagramm und andere Informationen über den von Ihnen ausgewählten mathematischen Ausdruck anzeigt.
11. Geben Sie für Wann immer der ***Ausdruck*** ist an, ob der Ausdruck größer, kleiner oder gleich dem Schwellenwert sein muss. Geben Sie unter than... (dann ...) den Schwellenwert an.
12. Wählen Sie Additional configuration (Zusätzliche Konfiguration). Geben Sie unter Datapoints to alarm (Datenpunkte für Alarm) an, wie viele Auswertungszeiträume (Datenpunkte) im Status `ALARM` sein müssen, damit der Alarm ausgelöst wird. Wenn die beiden Werte hier übereinstimmen, erstellen Sie einen Alarm, der in den Status `ALARM` wechselt, wenn entsprechend viele aufeinanderfolgende Zeiträume überschritten werden.

    Um einen M aus N Alarm zu erstellen, geben Sie eine niedrigere Zahl für den ersten Wert als für den zweiten Wert an. Weitere Informationen finden Sie unter Auswerten eines Alarms.
13. Wählen Sie für Missing data treatment (Behandlung von fehlenden Daten) aus, wie sich der Alarm verhalten soll, wenn einige Datenpunkte fehlen. Weitere Informationen finden Sie unter Konfiguration, wie Alarme fehlende Daten behandeln CloudWatch .
14. Wählen Sie Weiter.
15. Wählen Sie unter Notification (Benachrichtigung) ein SNS-Thema aus, das benachrichtigt werden soll, wenn sich der Alarm im Status `ALARM`, `OK` oder `INSUFFICIENT_DATA` befindet.

    Um zu erreichen, dass der Alarm mehrere Benachrichtigungen für den gleichen Alarmstatus oder für verschiedene Statuswerte sendet, wählen Sie Benachrichtigung hinzufügen.

    Damit der Alarm keine Benachrichtigungen sendet, wählen Sie Remove (Entfernen).
16. Um den Alarm Auto-Scaling-, EC2-, Lambda- oder Systems-Manager-Aktionen ausführen zu lassen, wählen Sie die entsprechende Schaltfläche und wählen Sie den Alarmstatus und die auszuführende Aktion. Wenn Sie eine Lambda-Funktion als Alarmaktion wählen, geben Sie den

Funktionsnamen oder ARN an, und Sie können optional eine bestimmte Version der Funktion auswählen.

Alarme können Aktionen des Systems Manager nur ausführen, wenn sie in den ALARM-Zustand wechseln. Weitere Informationen zu Systems Manager Manager-Aktionen finden Sie unter Konfiguration für CloudWatch die Erstellung OpsItems aus Alarmen und Incident-Erstellung.

> **Note**
>
> Um einen Alarm zu erstellen, der eine SSM-Incident-Manager-Aktion ausführt, müssen Sie über bestimmte Berechtigungen verfügen. Weitere Informationen finden Sie unter Beispiele für identitätsbasierte Richtlinien für AWS Systems Manager Incident Manager.

17. Wenn Sie fertig sind, wählen Sie Weiter.
18. Geben Sie einen Namen und eine Beschreibung für den Alarm ein. Wählen Sie anschließend Weiter.

    Der Name darf nur UTF-8-Zeichen und keine ASCII-Kontrolleingabezeichen enthalten. Die Beschreibung kann Markdown-Formatierungen enthalten, die nur auf der Registerkarte Alarmdetails in der Konsole angezeigt werden. CloudWatch Der Markdown kann nützlich sein, um Links zu Runbooks oder anderen internen Ressourcen hinzuzufügen.
19. Bestätigen Sie unter Preview and create (Vorschau und erstellen), dass die Informationen und Bedingungen den Anforderungen entsprechen, und wählen Sie dann Create alarm (Alarm erstellen).

Sie können auch Alarme zu einem Dashboard hinzufügen. Weitere Informationen finden Sie unter Fügen Sie ein Alarm-Widget zu einem CloudWatch Dashboard hinzu oder entfernen Sie es .

Erstellen Sie einen CloudWatch Alarm auf der Grundlage einer Metrics Insights-Abfrage

Sie können auch einen Alarm für alle Metrics-Insights-Abfragen einrichten, die eine einzelne Zeitreihe zurückgeben. Dies kann besonders nützlich sein, um Alarme zu erstellen, die aggregierte Metriken einer Flotte Ihrer Infrastruktur oder Ihrer Anwendungen überwachen. Wenn Sie den Alarm einmal erstellen, passt er sich dynamisch an, wenn Ressourcen zur Flotte hinzugefügt oder aus ihr entfernt werden. Sie können beispielsweise einen Alarm erstellen, der die CPU-Auslastung all Ihrer Instances überwacht, und der Alarm passt sich dynamisch an, wenn Sie Instances hinzufügen oder entfernen.

Vollständige Anweisungen finden Sie unter Alarme für Metrics-Insights-Abfragen erstellen.

Einen Alarm basierend auf einer verbundenen Datenquelle erstellen

Sie können Alarme erstellen, die Messwerte aus Datenquellen überwachen, die nicht vorhanden sind CloudWatch. Weitere Informationen zum Erstellen von Verbindungen zu diesen anderen Datenquellen finden Sie unter Metriken aus anderen Datenquellen abfragen.

So erstellen Sie einen Alarm für Metriken aus einer Datenquelle, mit der Sie eine Verbindung hergestellt haben

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken) All metrics (Alle Metriken) aus.
3. Wählen Sie die Registerkarte Abfrage mit mehreren Quellen.
4. Wählen Sie für Datenquelle die Datenquelle, die Sie verwenden möchten.
5. Der Abfragegenerator fordert Sie auf, die Informationen einzugeben, die für die Abfrage erforderlich sind, um die für den Alarm zu verwendenden Metriken abzurufen. Der Workflow ist für jede Datenquelle unterschiedlich und auf die Datenquelle zugeschnitten. Beispielsweise wird für Amazon Managed Service für Prometheus und Prometheus-Datenquellen ein PromQL-Abfrage-Editor-Feld mit einem Abfrage-Helfer angezeigt.
6. Wenn Sie mit der Erstellung der Abfrage fertig sind, wählen Sie Diagramm-Abfrage.
7. Wenn das Beispieldiagramm Ihren Erwartungen entspricht, wählen Sie Alarm erstellen.
8. Die Seite Metrik und Bedingungen angeben wird angezeigt. Wenn die Abfrage, die Sie verwenden, mehr als eine Zeitreihe produziert, sehen Sie oben auf der Seite eine Warnung. Wählen Sie in diesem Fall unter Aggregationsfunktion eine Funktion aus, mit der die Zeitreihe aggregiert werden soll.
9. (Optional) Fügen Sie ein Label für den Alarm hinzu.
10. Für wann immer ***your-metric-name***es ist. , wählen Sie Größer, Größer/Gleich, Niedriger/Gleich oder Niedriger. Geben Sie danach für als … eine Zahl für Ihren Schwellenwert an.
11. Wählen Sie Additional configuration (Zusätzliche Konfiguration). Geben Sie unter Datapoints to alarm (Datenpunkte für Alarm) an, wie viele Auswertungszeiträume (Datenpunkte) im Status `ALARM` sein müssen, damit der Alarm ausgelöst wird. Wenn die beiden Werte hier übereinstimmen, erstellen Sie einen Alarm, der in den Status `ALARM` wechselt, wenn entsprechend viele aufeinanderfolgende Zeiträume überschritten werden.

Um einen M-aus-N-Alarm zu erzeugen, geben Sie für den ersten Wert eine Zahl an, die niedriger ist als die Zahl für den zweiten Wert. Weitere Informationen finden Sie unter Auswerten eines Alarms.

12. Wählen Sie für Fehlende Datenverarbeitung aus, wie sich der Alarm verhalten soll, wenn einige Datenpunkte fehlen. Weitere Informationen finden Sie unter Konfiguration, wie Alarme fehlende Daten behandeln CloudWatch .
13. Wählen Sie Weiter.
14. Legen Sie für Benachrichtigung ein Amazon-SNS-Thema fest, das benachrichtigen soll, wenn der Alarm in den Status `ALARM`, `OK` oder `INSUFFICIENT_DATA` übergeht.

    a. (Optional) Um mehrere Benachrichtigungen für den gleichen Alarmstatus oder für verschiedene Statuswerte zu senden, wählen Sie Add notification (Benachrichtigung hinzufügen) aus.

    > **Note**
    >
    > Wir empfehlen, dass Sie den Alarm so einstellen, dass er zusätzlich zu dem Alarm-Zustand auch dann Maßnahmen ergreift, wenn er in den Status Ungenügend Daten wechselt. Dies liegt daran, dass viele Probleme mit der Lambda-Funktion, die eine Verbindung zur Datenquelle herstellt, dazu führen können, dass der Alarm auf Unzureichende Daten übergeht.

    b. (Optional) Damit keine Amazon-SNS-Benachrichtigungen gesendet werden, wählen Sie Entfernen.
15. Um den Alarm Auto-Scaling-, EC2-, Lambda- oder Systems-Manager-Aktionen ausführen zu lassen, wählen Sie die entsprechende Schaltfläche und wählen Sie den Alarmstatus und die auszuführende Aktion. Wenn Sie eine Lambda-Funktion als Alarmaktion wählen, geben Sie den Funktionsnamen oder ARN an, und Sie können optional eine bestimmte Version der Funktion auswählen.

    Alarme können Aktionen des Systems Manager nur ausführen, wenn sie in den ALARM-Zustand wechseln. Weitere Informationen zu Systems Manager Manager-Aktionen finden Sie unter Konfiguration für CloudWatch die Erstellung OpsItems aus Alarmen und Incident-Erstellung.

> **Note**
>
> Um einen Alarm zu erstellen, der eine SSM-Incident-Manager-Aktion ausführt, müssen Sie über bestimmte Berechtigungen verfügen. Weitere Informationen finden Sie unter Beispiele für identitätsbasierte Richtlinien für AWS Systems Manager Incident Manager.

16. Wählen Sie Weiter aus.
17. Geben Sie unter Name and description (Name und Beschreibung) einen Namen und eine Beschreibung für den Alarm ein und klicken Sie auf Next (Weiter). Der Name darf nur UTF-8-Zeichen und keine ASCII-Kontrolleingabezeichen enthalten. Die Beschreibung kann Markdown-Formatierungen enthalten, die nur auf der Registerkarte Alarmdetails in der Konsole angezeigt werden. CloudWatch Der Markdown kann nützlich sein, um Links zu Runbooks oder anderen internen Ressourcen hinzuzufügen.

    > **Tip**
    >
    > Der Alarmname darf nur UTF-8-Zeichen enthalten. Er darf keine ASCII-Steuerzeichen enthalten.

18. Bestätigen Sie unter Preview and create (Vorschau und erstellen), dass die Informationen und Bedingungen Ihres Alarms korrekt sind, und wählen Sie dann Create alarm (Alarm erstellen) aus.

Einzelheiten zu Alarmen für verbundene Datenquellen

- Wenn ein Alarm CloudWatch ausgewertet wird, erfolgt dies jede Minute, auch wenn der Zeitraum für den Alarm länger als eine Minute ist. Damit der Alarm funktioniert, muss die Lambda-Funktion in der Lage sein, eine Liste von Zeitstempeln zurückzugeben, die mit einer beliebigen Minute beginnen, nicht nur mit einem Vielfachen der Zeitraumlänge. Diese Zeitstempel müssen einen Abstand von einer Zeitraumlänge haben.

  Wenn die vom Lambda abgefragte Datenquelle daher nur Zeitstempel zurückgeben kann, die ein Vielfaches der Zeitraumlänge sind, sollte die Funktion die abgerufenen Daten „erneut abtasten“, damit sie den von der `GetMetricData`-Anfrage erwarteten Zeitstempeln entsprechen.

  Beispielsweise wird ein Alarm mit einem Zeitraum von fünf Minuten jede Minute anhand von Fünf-Minuten-Fenstern ausgewertet, die sich jedes Mal um eine Minute verschieben. In diesem Fall.

  - Für die Alarmauswertung um 12:15:00 Uhr werden Datenpunkte mit den Zeitstempeln, und CloudWatch erwartet. `12:00:00` `12:05:00` `12:10:00`
  - CloudWatch Erwartet dann für die Alarmauswertung um 12:16:00 Uhr Datenpunkte mit den Zeitstempeln, und. `12:01:00` `12:06:00` `12:11:00`
- Bei der CloudWatch Auswertung eines Alarms werden alle von der Lambda-Funktion zurückgegebenen Datenpunkte, die nicht mit den erwarteten Zeitstempeln übereinstimmen, gelöscht, und der Alarm wird anhand der verbleibenden erwarteten Datenpunkte ausgewertet. Wenn der Alarm beispielsweise bei der Auswertung ausgewertet wird, werden Daten mit den Zeitstempeln `12:15:00`, `12:00:00`, `12:05:00` und `12:10:00` erwartet. Wenn sie Daten mit den Zeitstempeln`12:00:00`,, und empfängt `12:05:0012:06:00`, werden die Daten von gelöscht und `12:10:00` der Alarm `12:06:00` wird anhand der anderen Zeitstempel CloudWatch ausgewertet.

  Für die nächste Auswertung um `12:16:00` werden dann Daten mit den Zeitstempeln `12:01:00`, `12:06:00` und `12:11:00` erwartet. Wenn nur die Daten mit den Zeitstempeln `12:00:00`, `12:05:00` und `12:10:00` vorliegen, werden all diese Datenpunkte um 12:16:00 Uhr ignoriert und der Alarm geht in den Zustand über, den Sie dem Alarm für die Behandlung fehlender Daten angegeben haben. Weitere Informationen finden Sie unter Auswerten eines Alarms.
- Es wird empfohlen, diese Alarme zu erstellen, um Maßnahmen zu ergreifen, wenn sie in den `INSUFFICIENT_DATA`-Zustand wechseln, da bei mehreren Anwendungsfällen mit Lambda-Funktionsausfällen der Alarm auf `INSUFFICIENT_DATA` wechselt, unabhängig davon, wie Sie den Alarm zur Behandlung fehlender Daten einstellen.
- Wenn die Lambda-Funktion einen Fehler oder unvollständige Daten zurückgibt:
  - Wenn beim Aufrufen der Lambda-Funktion ein Berechtigungsproblem auftritt, beginnt der Alarm mit fehlenden Datenübergängen, je nachdem, wie Sie den Alarm bei der Erstellung für die Behandlung fehlender Daten angegeben haben.
  - Wenn die Lambda-Funktion `'StatusCode' = 'PartialData'` zurückgibt, schlägt die Alarmauswertung fehl und der Alarm wechselt nach drei Versuchen auf `INSUFFICIENT_DATA`. Dies dauert etwa drei Minuten.
  - Jeder andere Fehler, der von der Lambda-Funktion kommt, führt dazu, dass der Alarm zu `INSUFFICIENT_DATA` wechselt.
- Wenn die von der Lambda-Funktion angeforderte Metrik eine gewisse Verzögerung aufweist, sodass der letzte Datenpunkt immer fehlt, sollten Sie eine Problemumgehung verwenden. Sie können einen M-aus-N-Alarm erstellen oder den Bewertungszeitraum des Alarms verlängern. Weitere Informationen über M-aus-N-Alarmen finden Sie unter Auswerten eines Alarms.

Erstellen Sie einen CloudWatch Alarm, der auf der Erkennung von Anomalien basiert

Sie können einen Alarm auf der Grundlage der CloudWatch Anomalieerkennung erstellen, der vergangene Metrikdaten analysiert und ein Modell der erwarteten Werte erstellt. Die erwarteten Werte berücksichtigen die typischen stündlichen, täglichen und wöchentlichen Muster in der Metrik.

Sie legen einen Wert für den Schwellenwert für die Erkennung von Anomalien fest und CloudWatch verwenden diesen Schwellenwert zusammen mit dem Modell, um den „normalen“ Wertebereich für die Metrik zu bestimmen. Ein höherer Wert für den Schwellenwert erzeugt ein breiteres Band „normaler“ Werte.

Sie können bestimmen, ob der Alarm ausgelöst werden soll, wenn der Metrikwert das Band erwarteter Werte überschreitet, das Band unterschreitet oder das Band über- oder unterschreitet.

Sie können auch Alarme für die Anomalieerkennung basierend auf einzelnen Metriken und Metrikberechnungs-Ausdrücken erstellen können. Sie können diese Ausdrücke verwenden, um Diagramme zu erstellen, die Anomalieerkennungsbänder visualisieren.

In einem Konto, das als Überwachungskonto für CloudWatch kontenübergreifende Beobachtbarkeit eingerichtet wurde, können Sie zusätzlich zu den Metriken im Überwachungskonto Anomaliedetektoren für Kennzahlen in Quellkonten erstellen.

Weitere Informationen finden Sie unter Verwendung der CloudWatch Anomalieerkennung.

> **Note**
>
> Wenn Sie in der Metriken-Konsole bereits Anomalieerkennung zu Darstellungszwecken bei einer Metrik einsetzen und Sie für ebendiese Metrik einen Alarm für die Anomalieerkennung erstellen, ändert der von Ihnen für den Alarm festgelegte Schwellenwert nicht den bereits für die Darstellung verwendeten Schwellenwert. Weitere Informationen finden Sie unter Erstellen eines Diagramms.

Gehen Sie zum Erstellen eines Alarms basierend auf der Anomalieerkennung wie folgt vor:

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Alarms (Alarme) und All alarms (Alle Alarme) aus.

3. Wählen Sie Create alarm (Alarm erstellen) aus.
4. Wählen Sie Select Metric (Metrik auswählen) aus.
5. Führen Sie eine der folgenden Aktionen aus:
   - Wählen Sie den Service-Namespace aus, der Ihre Metrik enthält, und wählen Sie dann weiterhin Optionen aus, so dass sie Ihre Optionen einschränken. Wenn eine Liste von Metriken angezeigt wird, aktivieren Sie das Kontrollkästchen neben der Metrik.
   - Geben Sie in das Suchfeld den Namen einer Metrik, Dimension oder Ressourcen-ID ein. Wählen Sie dann eines der Ergebnisse aus und wählen Sie weiterhin Optionen aus, bis eine Liste der Metriken erscheint. Aktivieren Sie das Kontrollkästchen neben der Metrik.
6. Wählen Sie Graphed Metric aus.

   a. (Optional) Wählen Sie für Statistik die Dropdownliste und wählen Sie dann eine der vordefinierten Statistiken oder Perzentile aus. Verwenden Sie das Suchfeld im Dropdown-Menü, um ein benutzerdefiniertes Perzentil anzugeben, wie z. B. **`p95.45`**.

   b. (Optional) Wählen Sie für Zeitraum die Dropdownliste aus und wählen Sie dann einen der vordefinierten Bewertungszeiträume aus.

   > **Note**
   >
   > Bei der CloudWatch Auswertung Ihres Alarms wird der Zeitraum zu einem einzigen Datenpunkt zusammengefasst. Für Anomalieerkennungsalarme muss der Auswertungszeitraum eine Minute oder länger sein.

7. Wählen Sie Weiter aus.
8. Geben Sie unter Conditions (Bedingungen) Folgendes an:

   a. Wählen Sie Anomaly detection (Anomalieerkennung).

   Wenn das Modell für diese Metrik und Statistik bereits vorhanden ist, CloudWatch wird im Diagramm oben auf dem Bildschirm eine Vorschau des Bandes zur Erkennung von Anomalien angezeigt. Nachdem Sie Ihren Alarm erstellt haben, kann es bis zu 15 Minuten dauern, bis das tatsächliche Anomalieerkennungsband im Diagramm angezeigt wird. Zuvor ist das angezeigte Band eine Approximation des Anomalieerkennungsbands.

> **Tip**
>
> Um das Diagramm oben auf dem Bildschirm für einen längeren Zeitraum anzuzeigen, wählen Sie oben rechts auf der Seite Edit (Bearbeiten) aus.

Wenn das Modell für diese Metrik und Statistik noch nicht vorhanden ist, wird das Band zur Erkennung von Anomalien CloudWatch generiert, nachdem Sie den Alarm erstellt haben. Bei neuen Modellen kann es bis zu drei Stunden dauern, bis das tatsächliche Anomalieerkennungsband im Diagramm erscheint. Es kann bis zu zwei Wochen dauern, bis das neue Modell trainiert ist, sodass das Erkennungsband für Anomalien genauere Erwartungswerte anzeigt.

b. Geben Sie bei Whenever ***`metric`*** is (Wenn Metrik … ist) an, wann der Alarm ausgelöst werden soll, etwa wenn die Metrik größer, niedriger oder außerhalb des Bereichs (in beide Richtungen) ist.

c. Wählen Sie in Anomaly detection threshold (Schwellenwert für die Anomalieerkennung) die Zahl aus, die für den Schwellenwert für die Anomalieerkennung verwendet werden soll. Eine höhere Zahl schafft ein breiteres Band "normaler" Werte, das toleranter gegenüber metrischen Änderungen ist. Eine niedrigere Zahl schafft ein dünneres Band, das mit geringeren metrischen Abweichungen in den `ALARM`-Status geht. Die Zahl muss nicht eine ganze Zahl sein.

d. Wählen Sie Additional configuration (Zusätzliche Konfiguration). Geben Sie unter Datapoints to alarm (Datenpunkte für Alarm) an, wie viele Auswertungszeiträume (Datenpunkte) im Status `ALARM` sein müssen, damit der Alarm ausgelöst wird. Wenn die beiden Werte hier übereinstimmen, erstellen Sie einen Alarm, der in den Status `ALARM` wechselt, wenn entsprechend viele aufeinanderfolgende Zeiträume überschritten werden.

   Um einen M-aus-N-Alarm zu erzeugen, geben Sie für den ersten Wert eine Zahl an, die niedriger ist als die Zahl für den zweiten Wert. Weitere Informationen finden Sie unter Auswerten eines Alarms.

e. Wählen Sie für Fehlende Datenverarbeitung aus, wie sich der Alarm verhalten soll, wenn einige Datenpunkte fehlen. Weitere Informationen finden Sie unter Konfiguration, wie Alarme fehlende Daten behandeln CloudWatch .

f. Wenn der Alarm ein Perzentil als überwachte Statistik verwendet, erscheint ein Feld Percentiles with low samples (Perzentile mit geringen Stichproben). Verwenden Sie diese

Option, um zu entscheiden, ob Sie Fälle mit niedrigem Stichprobenumfang bewerten oder ignorieren möchten. Wenn Sie Ignore (maintain alarm state) (Ignorieren (Alarmzustand beibehalten)) wählen, wird der aktuelle Alarmzustand immer beibehalten, wenn der Stichprobenumfang zu gering ist. Weitere Informationen finden Sie unter Auf Perzentilen basierende CloudWatch Alarme und Stichproben mit niedrigen Datenmengen.

9. Wählen Sie Weiter.
10. Wählen Sie unter Notification (Benachrichtigung) ein SNS-Thema aus, das benachrichtigt werden soll, wenn sich der Alarm im Status `ALARM`, `OK` oder `INSUFFICIENT_DATA` befindet.

    Um mehrere Benachrichtigungen für den gleichen Alarmstatus oder für verschiedene Statuswerte zu senden, wählen Sie Add notification (Benachrichtigung hinzufügen).

    Wählen Sie Remove (Entfernen), wenn Sie nicht möchten, dass der Alarm Benachrichtigungen sendet.
11. Sie können den Alarm so einrichten, dass er EC2-Aktionen ausführt oder eine Lambda-Funktion aufruft, wenn er seinen Status ändert, oder um einen Systems Manager OpsItem oder Incident zu erstellen, wenn er in den ALARM-Status wechselt. Wählen Sie dazu die entsprechende Schaltfläche und dann den Alarmzustand und die auszuführende Aktion aus.

    Wenn Sie eine Lambda-Funktion als Alarmaktion wählen, geben Sie den Funktionsnamen oder ARN an, und Sie können optional eine bestimmte Version der Funktion auswählen.

    Weitere Informationen zu Systems Manager Manager-Aktionen finden Sie unter Konfiguration für CloudWatch die Erstellung OpsItems aus Alarmen und Incident-Erstellung.

    > **Note**
    >
    > Um einen Alarm zu erstellen, der eine AWS Systems Manager -Incident-Manager-Aktion ausführt, müssen Sie über bestimmte Berechtigungen verfügen. Weitere Informationen finden Sie unter Beispiele für identitätsbasierte Richtlinien für AWS Systems Manager Incident Manager.
12. Wählen Sie Weiter aus.
13. Geben Sie unter Name and description (Name und Beschreibung) einen Namen und eine Beschreibung für den Alarm ein und klicken Sie auf Next (Weiter). Der Name darf nur UTF-8-Zeichen und keine ASCII-Kontrolleingabezeichen enthalten. Die Beschreibung kann Markdown-Formatierungen enthalten, die nur auf der Registerkarte Alarmdetails in der Konsole angezeigt

werden. CloudWatch Der Markdown kann nützlich sein, um Links zu Runbooks oder anderen internen Ressourcen hinzuzufügen.

> **Tip**
>
> Der Alarmname darf nur UTF-8-Zeichen und keine ASCII-Kontrolleingabezeichen enthalten

14. Bestätigen Sie unter Preview and create (Vorschau und erstellen), dass die Informationen und Bedingungen Ihres Alarms korrekt sind, und wählen Sie dann Create alarm (Alarm erstellen) aus.

Ändern eines Anomalieerkennungsmodells

Sobald Sie einen Alarm erstellt haben, können Sie das Anomalieerkennungsmodell anpassen. Sie können bestimmte Zeiträume von der Verwendung in der Modellerstellung ausschließen. Es ist wichtig, dass Sie ungewöhnliche Ereignisse wie Systemausfälle, Bereitstellungen und Feiertage von den Trainingsdaten ausschließen. Sie können auch angeben, ob das Modell an Zeitumstellungen (Sommer- und Winterzeit) angepasst werden soll.

So passen Sie das Anomalieerkennungsmodell für einen Alarm an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Alarms (Alarme) und All alarms (Alle Alarme) aus.
3. Wählen Sie den Namen des Alarms. Verwenden Sie ggf. das Suchfeld, um den Alarm zu finden.
4. Wählen Sie Analysieren, In Metriken aus.
5. Wählen Sie in der Spalte Details die Option ANOMALY_DETECTION_BAND, Anomalieerkennungsmodell bearbeiten aus.
6. Um einen Zeitraum von der Erstellung des Modells auszuschließen, wählen Sie das Kalendersymbol unter Enddatum. Wählen Sie dann die Tage und Uhrzeiten aus, die vom Training ausgeschlossen werden sollen, und wählen Sie Apply (Anwenden).
7. Wenn die Metrik gegenüber der Umstellung auf die Sommerzeit empfindlich ist, wählen Sie im Feld metric timezone (Metrik-Zeitzone) die entsprechende Zeitzone aus.
8. Wählen Sie Aktualisieren.

Löschen eines Anomalieerkennungsmodells

Bei der Nutzung der Anomalieerkennung für einen Alarm fallen -Gebühren an. Wenn Ihr Alarm kein Anomalie-Dektionsmodell mehr benötigt, ist es eine bewährte Methode, zuerst den Alarm und erst dann das Modell zu löschen. Wenn Anomalie-Dektionsalarme ausgewertet werden, werden alle fehlenden Anomaliedetektoren in Ihrem Namen erstellt. Wenn Sie das Modell löschen, ohne den Alarm zu löschen, wird das Modell vom Alarm automatisch erneut generiert.

So löschen Sie einen Alarm

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich zuerst Alarme und dann Alle Alarme aus.
3. Wählen Sie den Namen des Alarms.
4. Wählen Sie Aktionen, Löschen aus.
5. Wählen Sie im Bestätigungsfeld Löschen aus.

Das Anomalieerkennungsmodell löschen, das für einen Alarm verwendet wurde

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metrics (Metriken) und dann All metrics (Alle Metriken) aus.
3. Wählen Sie Browse (Durchsuchen) und dann die Metrik aus, die das Anomalieerkennungsmodell enthält. Sie können im Suchfeld nach Ihrer Metrik suchen oder Ihre Metrik über die Optionen auswählen.
   - (Optional) Wenn Sie die ursprüngliche Benutzeroberfläche verwenden, wählen Sie All Metrics (Alle Metriken) und dann die Metrik aus, die das Anomalieerkennungsmodell enthält. Sie können im Suchfeld nach Ihrer Metrik suchen oder Ihre Metrik über die Optionen auswählen.
4. Wählen Sie Graphed metrics (Grafisch dargestellte Metrik) aus.
5. Auf der Registerkarte Graphed metrics (Grafisch dargestellte Metriken) wählen Sie den Namen des Anomalieerkennungsmodells aus, das Sie entfernen möchten, und wählen Sie dann Delete anomaly detection model (Anomalieerkennungsmodell löschen) aus.
   - (Optional) Wenn Sie die ursprüngliche Benutzeroberfläche verwenden, wählen Sie Edit model (Modell bearbeiten) aus. Sie werden zu einem neuen Bildschirm weitergeleitet. Wählen Sie auf dem neuen Bildschirm Delete model (Modell löschen) und dann Delete (Löschen) aus.

Alarmieren in Protokollen

In den folgenden Abschnitten wird erläutert, wie CloudWatch Alarme in Protokollen erstellt werden.

Erstellen Sie einen CloudWatch Alarm auf der Grundlage eines Metrikfilters für Protokollgruppen

Das Verfahren in diesem Abschnitt beschreibt, wie ein Alarm basierend auf einem Protokollgruppen-Metrikfilter erstellt wird. Mit metrischen Filtern können Sie in Protokolldaten, an die die Daten gesendet werden, nach Begriffen und Mustern suchen. CloudWatch Weitere Informationen finden Sie unter Metriken aus Protokollereignissen mithilfe von Filtern erstellen im Amazon CloudWatch Logs-Benutzerhandbuch. Bevor Sie einen Alarm auf der Grundlage eines Protokollgruppen-Metrikfilters erstellen, müssen Sie die folgenden Aktionen ausführen:

- Erstellen Sie eine -Protokollgruppe. Weitere Informationen finden Sie unter Arbeiten mit Protokollgruppen und Protokollstreams im Amazon CloudWatch Logs-Benutzerhandbuch.
- Erstellen Sie einen Metrikfilter. Weitere Informationen finden Sie unter Erstellen eines Metrikfilters für eine Protokollgruppe im Amazon CloudWatch Logs-Benutzerhandbuch.

Erstellen eines Alarms basierend auf einem Protokollgruppen-Metrikfilter

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie ausgehend vom Navigationsbereich Logs (Protokolle) und dann Log groups (Protokollgruppen) aus.
3. Wählen Sie die Protokollgruppe aus, die Ihren Metrikfilter enthält.
4. Wählen Sie Metric filters (Metrikfilter) aus.
5. Wählen Sie in der Registerkarte „Metric filters“ (Metrikfilter) das Feld für den Metrikfilter aus, auf dem Ihr Alarm basieren soll.
6. Wählen Sie Alarm erstellen aus.
7. (Optional) Unter Metric (Metrik) können Sie Metric name (Metrikname), Statistic (Statistik) und Period (Zeitraum) bearbeiten.
8. Geben Sie unter Conditions (Bedingungen) Folgendes an:

    a. Wählen Sie als Threshold type (Schwellenwerttyp) die Option Static (Statisch) oder Anomaly detection (Anomalieerkennung) aus.

b. Für wann immer ***your-metric-name***es ist. , wählen Sie Größer, Größer/Gleich, Niedriger/Gleich oder Niedriger.

c. Für than . . . (als . . .) geben Sie eine Zahl für Ihren Schwellenwert an.

9. Wählen Sie Additional configuration (Zusätzliche Konfiguration).

   a. Geben Sie für Data points to Alarm (Datenpunkte für den Alarm) an, wie viele Datenpunkte Ihren Alarm dazu bringen, in den `ALARM`-Status überzugehen. Wenn Sie übereinstimmende Werte angeben, wechselt Ihr Alarm in den Status `ALARM`, wenn entsprechend viele aufeinanderfolgende Zeiträume überschritten werden. Um einen M-aus-N-Alarm zu erzeugen, geben Sie für den ersten Wert eine Zahl an, die niedriger als die Zahl ist, die Sie für den zweiten Wert angeben. Weitere Informationen finden Sie unter CloudWatch Amazon-Alarme verwenden.

   b. Wählen Sie für Missing data treatment (Behandlung fehlender Daten) eine Option aus, um anzugeben, wie bei der Bewertung Ihres Alarms fehlende Daten zu behandeln sind.

10. Wählen Sie Weiter aus.

11. Legen Sie für Notification (Benachrichtigung) ein Amazon-SNS-Thema fest, das benachrichtigen soll, wenn sich der Alarm im Status `ALARM`, `OK` oder `INSUFFICIENT_DATA` befindet.

    a. (Optional) Um mehrere Benachrichtigungen für den gleichen Alarmstatus oder für verschiedene Statuswerte zu senden, wählen Sie Add notification (Benachrichtigung hinzufügen) aus.

    b. (Optional) Damit keine Benachrichtigungen gesendet werden, wählen Sie Remove (Entfernen) aus.

12. Um den Alarm Auto-Scaling-, EC2-, Lambda- oder Systems-Manager-Aktionen ausführen zu lassen, wählen Sie die entsprechende Schaltfläche und wählen Sie den Alarmstatus und die auszuführende Aktion. Wenn Sie eine Lambda-Funktion als Alarmaktion wählen, geben Sie den Funktionsnamen oder ARN an, und Sie können optional eine bestimmte Version der Funktion auswählen.

    Alarme können Aktionen des Systems Manager nur ausführen, wenn sie in den ALARM-Zustand wechseln. Weitere Informationen zu Systems Manager Manager-Aktionen finden Sie unter Konfiguration für CloudWatch die Erstellung OpsItems aus Alarmen und Incident-Erstellung.

> **Note**
>
> Um einen Alarm zu erstellen, der eine SSM-Incident-Manager-Aktion ausführt, müssen Sie über bestimmte Berechtigungen verfügen. Weitere Informationen finden Sie unter Beispiele für identitätsbasierte Richtlinien für AWS Systems Manager Incident Manager.

13. Wählen Sie Weiter aus.
14. Geben Sie unter Name und Beschreibung einen Namen und eine Beschreibung für Ihren Alarm ein. Der Name darf nur UTF-8-Zeichen und keine ASCII-Kontrolleingabezeichen enthalten. Die Beschreibung kann Markdown-Formatierungen enthalten, die nur auf der Registerkarte Alarmdetails in der Konsole angezeigt werden. CloudWatch Der Markdown kann nützlich sein, um Links zu Runbooks oder anderen internen Ressourcen hinzuzufügen.
15. Vergewissern Sie sich für Preview and create (Vorschau anzeigen und erstellen), ob Ihre Konfiguration korrekt ist, und wählen Sie Create alarm (Alarm erstellen) aus.

Kombinieren von Alarmen

Mit CloudWatch können Sie mehrere Alarme zu einem zusammengesetzten Alarm kombinieren, um einen zusammengefassten, aggregierten Statusindikator für eine ganze Anwendung oder Gruppe von Ressourcen zu erstellen. Verbundalarme sind Alarme, die ihren Zustand durch die Überwachung der Zustände anderer Alarme bestimmen. Sie definieren Regeln, um den Status dieser überwachten Alarme mithilfe einer booleschen Logik zu kombinieren.

Sie können Verbundalarme verwenden, um Alarmrauschen zu reduzieren, indem Sie Aktionen nur auf aggregierter Ebene durchführen. Sie können beispielsweise einen Verbundalarm erstellen, um eine Benachrichtigung an Ihr Webserver-Team zu senden, wenn ein Alarm im Zusammenhang mit Ihrem Webserver ausgelöst wird. Wenn einer dieser Alarme in den ALARM-Status übergeht, wechselt der Verbundalarm selbst in den ALARM-Status und sendet eine Benachrichtigung an Ihr Team. Wenn andere Alarme, die sich auf Ihren Webserver beziehen, ebenfalls in den ALARM-Status wechseln, wird Ihr Team nicht mit neuen Benachrichtigungen überlastet, da der Verbundalarm sie bereits über die aktuelle Situation informiert hat.

Sie können auch Verbundalarme verwenden, um komplexe Alarmbedingungen zu erstellen und nur dann Maßnahmen zu ergreifen, wenn viele verschiedene Bedingungen erfüllt sind. Sie können beispielsweise einen Verbundalarm erstellen, der einen CPU-Alarm und einen Speicheralarm

kombiniert und Ihr Team nur benachrichtigt, wenn sowohl der CPU- als auch der Speicheralarm ausgelöst wurden.

Using composite alarms

Bei der Verwendung von Verbundalarmen haben Sie zwei Möglichkeiten:

- Konfigurieren Sie die gewünschten Aktionen nur auf der Ebene des Verbundalarms, und erstellen Sie die zugrunde liegenden überwachten Alarme ohne Aktionen
- Konfigurieren Sie einen anderen Satz von Aktionen auf der Ebene des Verbundalarms. Bei den Aktionen für Verbundalarme könnte beispielsweise ein anderes Team involviert werden, falls ein weit verbreitetes Problem auftritt.

Verbundalarme können nur folgende Aktionen ausführen:

- Benachrichtigen von Amazon-SNS-Themen
- Aufrufen von Lambda-Funktionen
- OpsItems Im Systems Manager Ops Center erstellen
- Erstellen von Vorfällen in Systems Manager Incident Manager

> **Note**
>
> Alle zugrunde liegenden Alarme Ihres Verbundalarms müssen sich unter dem gleichen Konto und in der gleichen Region befinden wie Ihr Verbundalarm befinden. Wenn Sie jedoch einen zusammengesetzten Alarm in einem CloudWatch Konto für die kontenübergreifende Überwachung der Beobachtbarkeit einrichten, können die zugrunde liegenden Alarme Metriken in verschiedenen Quellkonten und im Überwachungskonto selbst überwachen. Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit. Ein einzelner zusammengesetzter Alarm kann 100 zugrunde liegende Alarme überwachen und 150 zusammengesetzte Alarme können einen einzelnen zugrunde liegenden Alarm überwachen.

Regelausdrücke

Alle zusammengesetzten Alarme enthalten Regelausdrücke. Über Regelausdrücke wird zusammengesetzten Alarmen mitgeteilt, welche anderen Alarme überwacht werden sollen, um ihre

Zustände zu bestimmen. Regelausdrücke können sich auf Metrikalarme und auf zusammengesetzte Alarme beziehen. Wenn Sie in einem Regelausdruck auf einen Alarm verweisen, weisen Sie dem Alarm eine Funktion zu, die bestimmt, in welchem der folgenden drei Zustände sich der Alarm befindet:

- ALARM

  ALARM („Alarmname oder Alarm-ARN“) ist TRUE, wenn sich der Alarm im ALARM-Status befindet.
- OK

  OK („Alarmname oder Alarm-ARN“) ist TRUE, wenn sich der Alarm im OK-Status befindet.
- INSUFFICIENT_DATA

  INSUFFICIENT_DATA („Alarmname oder Alarm-ARN“) ist TRUE, wenn sich der Alarm im INSUFFICIENT_DATA-Status befindet.

> **Note**
>
> TRUE wird immer als TRUE und FALSE immer als FALSE ausgewertet.

Beispielausdrücke

Der Anforderungsparameter `AlarmRule` unterstützt die Verwendung der logischen Operatoren `AND`, `OR` und `NOT`, sodass Sie mehrere Funktionen zu einem einzelnen Ausdruck kombinieren können. Die folgenden Beispielausdrücke zeigen, wie Sie die zugrunde liegenden Alarme in Ihrem zusammengesetzten Alarm konfigurieren können:

- `ALARM(CPUUtilizationTooHigh) AND ALARM(DiskReadOpsTooHigh)`

  Der Ausdruck gibt an, dass der zusammengesetzte Alarm nur in den Zustand `ALARM` wechselt, wenn sich `CPUUtilizationTooHigh` und `DiskReadOpsTooHigh` im Zustand `ALARM` befinden.
- `ALARM(CPUUtilizationTooHigh) AND NOT ALARM(DeploymentInProgress)`

  Der Ausdruck gibt an, dass der zusammengesetzte Alarm in den Zustand `ALARM` wechselt, wenn `CPUUtilizationTooHigh` sich im Zustand `ALARM` und `DeploymentInProgress` sich nicht im Zustand `ALARM` befindet. Dies ist ein Beispiel für einen zusammengesetzten Alarm, der Alarmrauschen während eines Bereitstellungsfensters reduziert.

- `(ALARM(CPUUtilizationTooHigh) OR ALARM(DiskReadOpsTooHigh)) AND OK(NetworkOutTooHigh)`

  Der Ausdruck gibt an, dass der zusammengesetzte Alarm in den Zustand `ALARM` wechselt, wenn `(ALARM(CPUUtilizationTooHigh)` oder `(DiskReadOpsTooHigh)` sich im Zustand `ALARM` und `(NetworkOutTooHigh)` sich im Zustand `OK` befindet. Dies ist ein Beispiel für einen zusammengesetzten Alarm, der Alarmrauschen reduziert, indem er Ihnen keine Benachrichtigungen sendet, wenn sich einer der zugrunde liegenden Alarme während eines Netzwerkproblems nicht im Zustand `ALARM` befindet.

Themen

Einen zusammengesetzten Alarm erstellen

In den Schritten in diesem Abschnitt wird erklärt, wie Sie mithilfe der CloudWatch Konsole einen zusammengesetzten Alarm erstellen. Sie können auch die API verwenden oder AWS CLI einen zusammengesetzten Alarm erstellen. Weitere Informationen finden Sie unter PutCompositeAlarmoder put-composite-alarm

So erstellen Sie einen zusammengesetzten Alarm

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Alarme und dann Alle Alarme aus.
3. Aktivieren Sie in der Liste der Alarme die Kontrollkästchen der vorhandenen Alarme, auf die Sie in Ihrem Regelausdruck verweisen möchten, und wählen Sie anschließend Create composite alarm (Zusammengesetzten Alarm erstellen) aus.
4. Geben Sie unter Specify composite alarm conditions (Festlegen von Bedingungen für zusammengesetzte Alarme) den Regelausdruck für den neuen zusammengesetzten Alarm an.

> **Note**
>
> Die Alarme, die Sie in der Liste der Alarme ausgewählt haben, werden automatisch im Feld Conditions (Bedingungen) aufgeführt. Standardmäßig wurde für jeden Ihrer Alarme

die Funktion ALARM festgelegt und die Alarme sind jeweils durch den logischen Operator OR verknüpft.

Der Regelausdruck kann mithilfe der folgenden Teilschritte geändert werden:

a. Sie können den erforderlichen Zustand für jeden einzelnen Ihrer Alarme von ALARM in OK oder INSUFFICENT_DATA ändern.
b. Sie können den logischen Operator in Ihrem Regelausdruck von OR in AND oder NOT ändern und Klammern hinzufügen, um Ihre Funktionen zu gruppieren.
c. Sie können andere Alarme in Ihren Regelausdruck einschließen oder Alarme aus Ihrem Regelausdruck löschen.

Beispiel: Regelausdruck mit Bedingungen

```
(ALARM("CPUUtilizationTooHigh") OR
ALARM("DiskReadOpsTooHigh")) AND
OK("NetworkOutTooHigh")
```

Im Beispiel eines Regelausdrucks, bei dem der zusammengesetzte Alarm ausgelöst wird, ALARM wenn ALARM (UtilizationTooHighDiskReadOpsTooHigh„CPU“ oder ALARM (“ „) aktiviert ist und gleichzeitig OK (“ NetworkOutTooHigh „) aktiviert istOK. ALARM

5. Wenn Sie fertig sind, wählen Sie Weiter.
6. Unter Configure actions (Aktionen konfigurieren) stehen folgende Optionen zur Auswahl:

   Für Notification (Benachrichtigung)

   - Select an exisiting SNS topic (Ein vorhandenes SNS-Thema auswählen), Create a new SNS topic (Ein neues SNS-Thema erstellen) oder Use a topic ARN (Einen Themen-ARN verwenden), um das SNS-Thema zu definieren, das die Benachrichtigung erhalten soll.
   - Add notification (Benachrichtigung hinzufügen), damit Ihr Alarm mehrere Benachrichtigungen für den gleichen Alarmzustand oder für verschiedene Alarmzustände senden kann.
   - Remove (Entfernen), damit von Ihrem Alarm keine Benachrichtigungen gesendet und keine Aktionen ausgeführt werden.

(Optional) Damit der Alarm eine Lambda-Funktion aufruft, wenn er seinen Status ändert, wählen Sie Lambda-Aktion hinzufügen. Geben Sie dann den Funktionsnamen oder den ARN an und wählen Sie optional eine bestimmte Version der Funktion aus.

Für Systems-Manager-Aktion

- Add Systems Manager action (Systems-Manager-Aktion hinzufügen), damit Ihr Alarm eine SSM-Aktion ausführen kann, wenn er in den ALARM-Zustand wechselt.

Weitere Informationen zu Systems Manager Manager-Aktionen finden Sie unter Konfiguration CloudWatch für die Erstellung OpsItems aus Alarmen im AWS Systems Manager Benutzerhandbuch und Incident-Erstellung im Incident Manager-Benutzerhandbuch. Um einen Alarm zu erstellen, der eine SSM-Incident-Manager-Aktion ausführt, müssen Sie über die richtigen Berechtigungen verfügen. Weitere Informationen finden Sie unter Beispiele für identitätsbasierte Richtlinien für AWS Systems Manager Incident Manager im Incident Manager-Benutzerhandbuch.

7. Wenn Sie fertig sind, wählen Sie Weiter.
8. Geben Sie unter Add name and description (Name und Beschreibung hinzufügen) einen Alarmnamen und optional eine Beschreibung für Ihren neuen zusammengesetzten Alarm ein. Der Name darf nur UTF-8-Zeichen und keine ASCII-Kontrolleingabezeichen enthalten. Die Beschreibung kann Markdown-Formatierungen enthalten, die nur auf der Registerkarte „Alarmdetails“ in der Konsole angezeigt werden. CloudWatch Der Markdown kann nützlich sein, um Links zu Runbooks oder anderen internen Ressourcen hinzuzufügen.
9. Wenn Sie fertig sind, wählen Sie Weiter.
10. Bestätigen Sie unter Preview and create (Vorschau und Erstellung) Ihre Angaben und wählen Sie anschließend Create composite alarm (Zusammengesetzten Alarm erstellen) aus.

> **Note**
>
> Es kann vorkommen, dass Sie einen Zyklus von zusammengesetzten Alarmen erstellen, bei dem zwei zusammengesetzte Alarme voneinander abhängen. In einem solchen Szenario werden Ihre zusammengesetzten Alarme nicht mehr ausgewertet und Sie können Ihre zusammengesetzten Alarme nicht mehr löschen, da sie voneinander abhängig sind. Der einfachste Weg, den Abhängigkeitszyklus zwischen Ihren

zusammengesetzten Alarmen zu durchbrechen, besteht darin, die Funktion `AlarmRule` in einem Ihrer zusammengesetzten Alarme in `False` zu ändern.

Unterdrücken von Verbundalarm-Aktionen

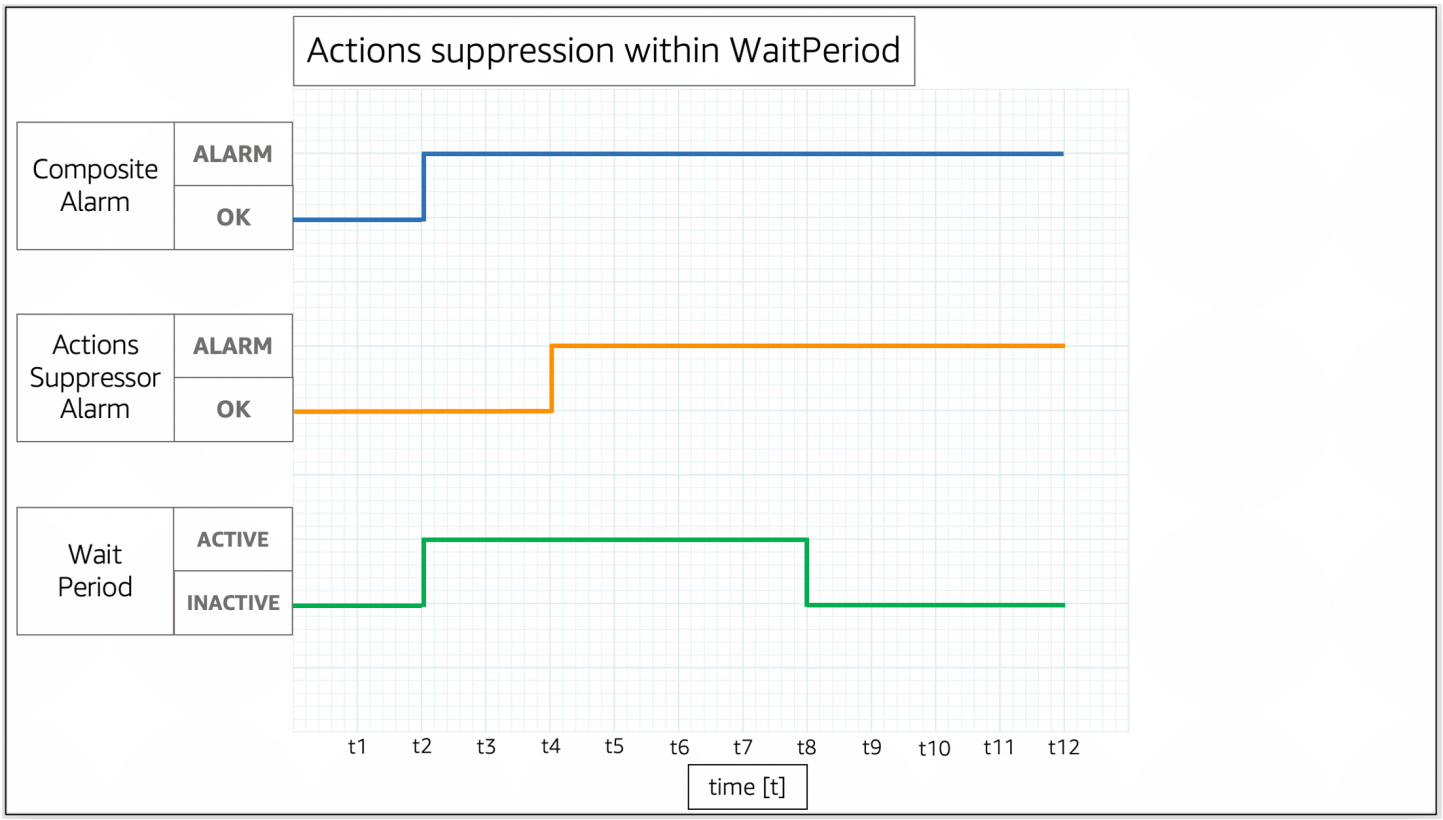
Da Verbundalarme es Ihnen ermöglichen, einen aggregierten Überblick über Ihren Zustand bei mehreren Alarmen zu erhalten, gibt es gängige Situationen, in denen erwartet wird, dass diese Alarme ausgelöst werden. Zum Beispiel während eines Wartungsfensters Ihrer Anwendung oder wenn Sie einen laufenden Vorfall untersuchen. In solchen Situationen möchten Sie möglicherweise die Aktionen Ihrer Verbundalarme unterdrücken, um unerwünschte Benachrichtigungen oder die Erstellung neuer Vorfall-Tickets zu verhindern

Mit der Aktionsunterdrückung für zusammengesetzte Alarme können Sie Alarme als Unterdrückungsalarme definieren. Unterdrückungsalarme verhindern, dass zusammengesetzte Alarme Aktionen ausführen. Sie können beispielsweise einen Unterdrückungsalarm angeben, der den Status einer unterstützenden Ressource darstellt. Wenn die unterstützende Ressource ausgefallen ist, verhindert der Unterdrückungsalarm, dass der zusammengesetzte Alarm Benachrichtigungen sendet. Die Aktionsunterdrückung für zusammengesetzte Alarme trägt zur Reduzierung von Alarmrauschen bei, sodass Sie weniger Zeit mit der Verwaltung Ihrer Alarme verbringen müssen und sich besser auf Ihre Abläufe konzentrieren können.

Unterdrückungsalarme werden bei der Konfiguration zusammengesetzter Alarme angegeben. Jeder Alarm kann als Unterdrückungsalarm fungieren. Wenn sich der Zustand eines Unterdrückungsalarms von `OK` in `ALARM` ändert, werden von dem zugehörigen zusammengesetzten Alarm keine Aktionen mehr ausgeführt. Wenn sich der Zustand eines Unterdrückungsalarms von `ALARM` in `OK` ändert, führt der zugehörige zusammengesetzte Alarm wieder Aktionen aus.

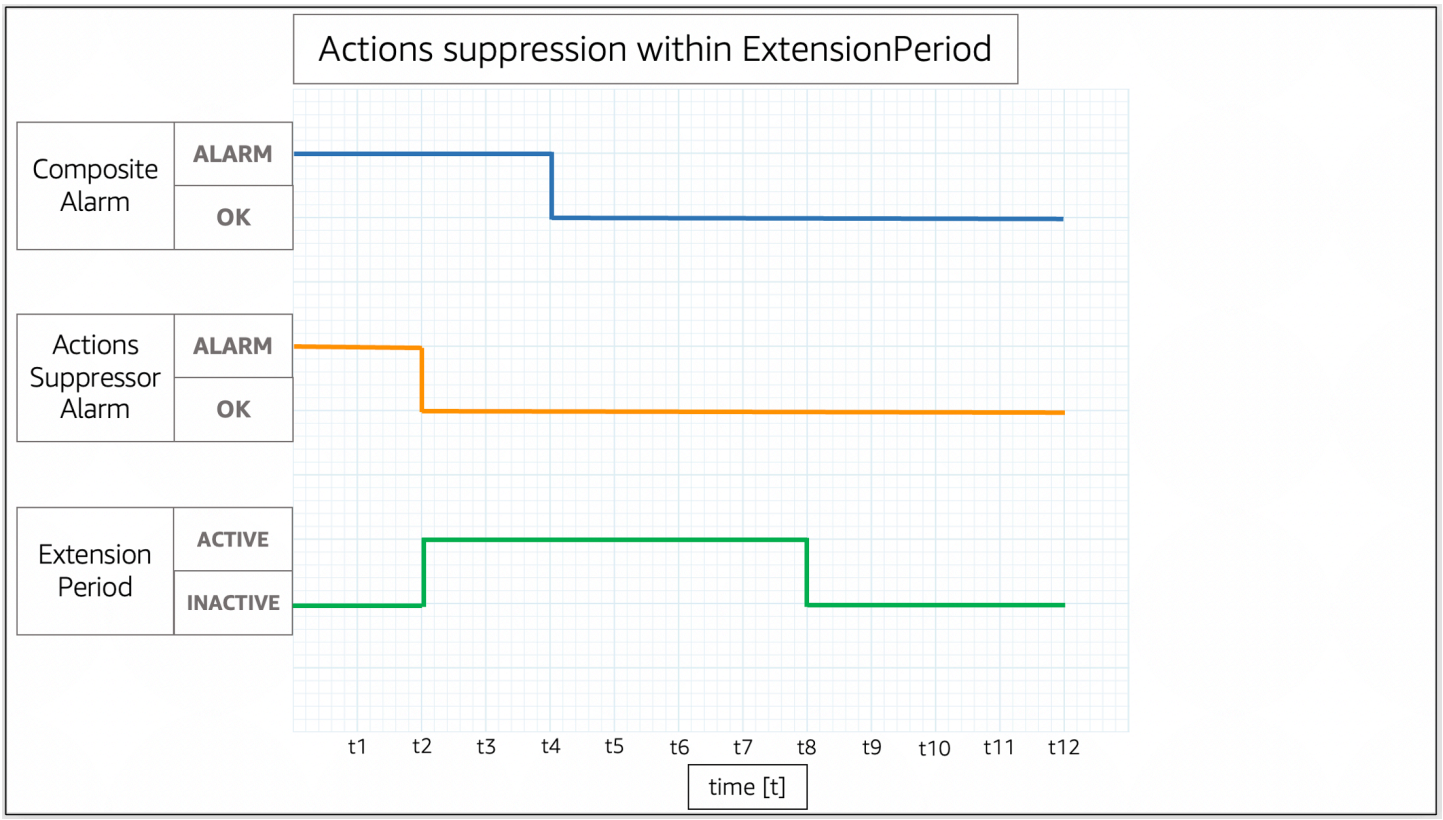
`WaitPeriod` und **`ExtensionPeriod`**

Wenn Sie einen Unterdrückungsalarm angeben, legen Sie die Parameter `WaitPeriod` und `ExtensionPeriod` fest. Diese Parameter verhindern, dass zusammengesetzte Alarme während der Zustandsänderung von Unterdrückungsalarmen unerwartet Aktionen ausführen. Verwenden Sie `WaitPeriod`, um Verzögerungen auszugleichen, die auftreten können, wenn sich der Zustand eines Unterdrückungsalarms von `OK` in `ALARM` ändert. Wenn sich also beispielsweise der Zustand eines Unterdrückungsalarms innerhalb von 60 Sekunden von `OK` in `ALARM` ändert, legen Sie `WaitPeriod` auf 60 Sekunden fest.



Im Image ändert sich der Zustand des zusammengesetzten Alarms bei t2 von OK in ALARM. Eine Wartezeit (`WaitPeriod`) beginnt bei t2 und endet bei t8. Dadurch hat der Unterdrückungsalarm Zeit, den Zustand bei t4 von OK in ALARM zu ändern, bevor die Aktionen des zusammengesetzten Alarms unterdrückt werden, wenn die Wartezeit (`WaitPeriod`) bei t8 abläuft.

Verwenden Sie `ExtensionPeriod`, um Verzögerungen auszugleichen, die auftreten können, wenn sich der Zustand eines zusammengesetzten Alarms in OK ändert, nachdem sich der Zustand eines Unterdrückungsalarms in OK geändert hat. Wenn sich also beispielsweise der Zustand eines Unterdrückungsalarms in OK geändert hat und sich der Zustand eines zusammengesetzten Alarms danach innerhalb von 60 Sekunden in OK ändert, legen Sie `ExtensionPeriod` auf 60 Sekunden fest.



Im Image ändert sich der Zustand des Unterdrückungsalarms bei t2 von `ALARM` in `OK`. Ein Verlängerungszeitraum (`ExtensionPeriod`) beginnt bei t2 und endet bei t8. Dadurch hat der zusammengesetzte Alarm Zeit, seinen Zustand von `ALARM` in `OK` zu ändern, bevor der Verlängerungszeitraum (`ExtensionPeriod`) bei t8 abläuft.

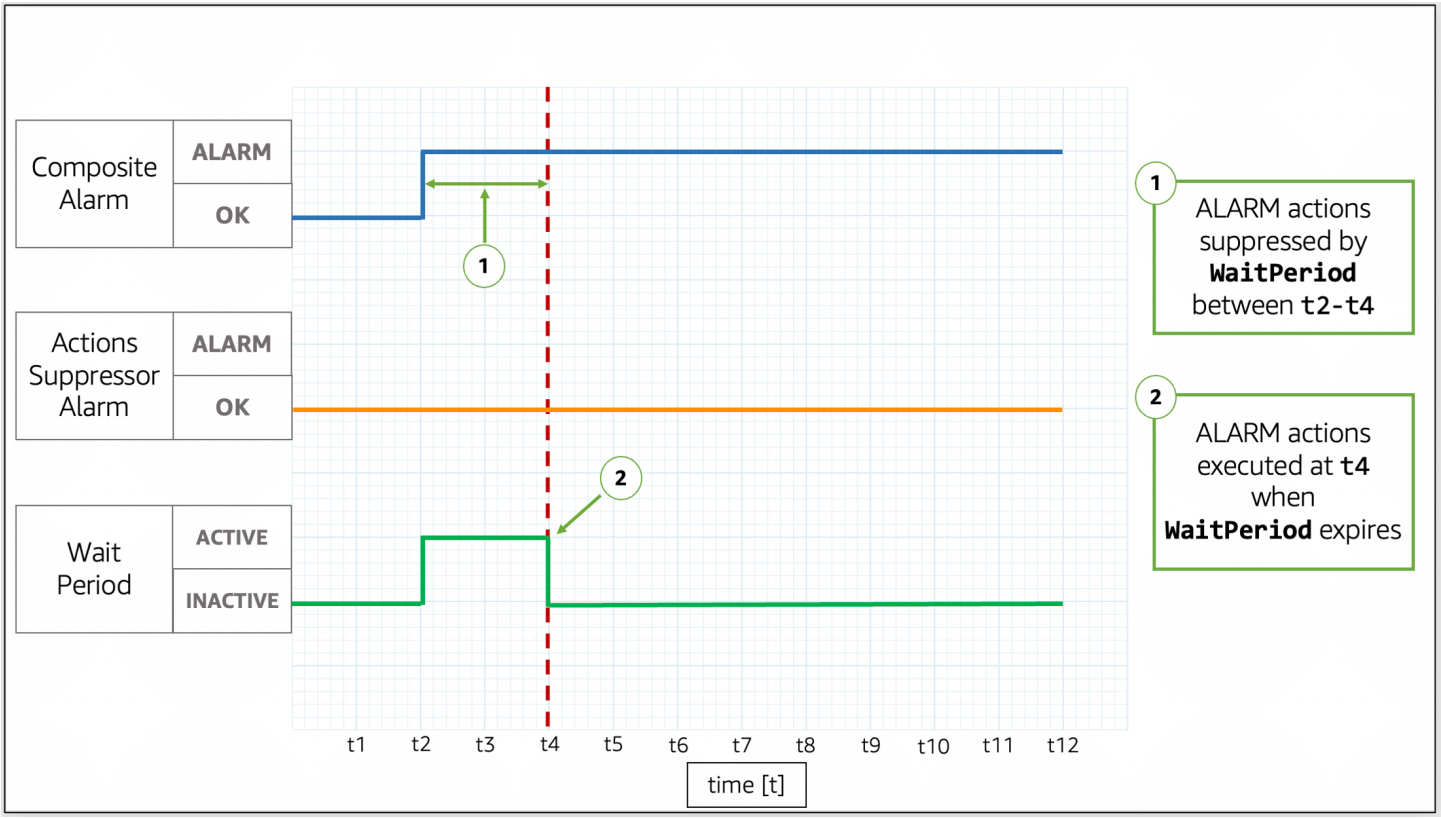
Von zusammengesetzten Alarmen werden keine Aktionen ausgeführt, wenn `WaitPeriod` und `ExtensionPeriod` aktiv werden. Zusammengesetzte Alarme führen Aktionen aus, die auf ihrem aktuellen Zustand basieren, wenn`ExtensionPeriod` und `WaitPeriod` inaktiv werden. Es wird empfohlen, den Wert für jeden Parameter auf 60 Sekunden festzulegen, da metrische Alarme jede Minute CloudWatch ausgewertet werden. Die Parameter können auf einen beliebigen ganzzahligen Sekundenwert festgelegt werden.

In den folgenden Beispielen wird detaillierter beschrieben, wie `WaitPeriod` und `ExtensionPeriod` verhindern, dass zusammengesetzte Alarme unerwartet Aktionen ausführen.

> **Note**
>
> In den folgenden Beispielen wird `WaitPeriod` als 2 Zeiteinheiten und `ExtensionPeriod` als 3 Zeiteinheiten konfiguriert.

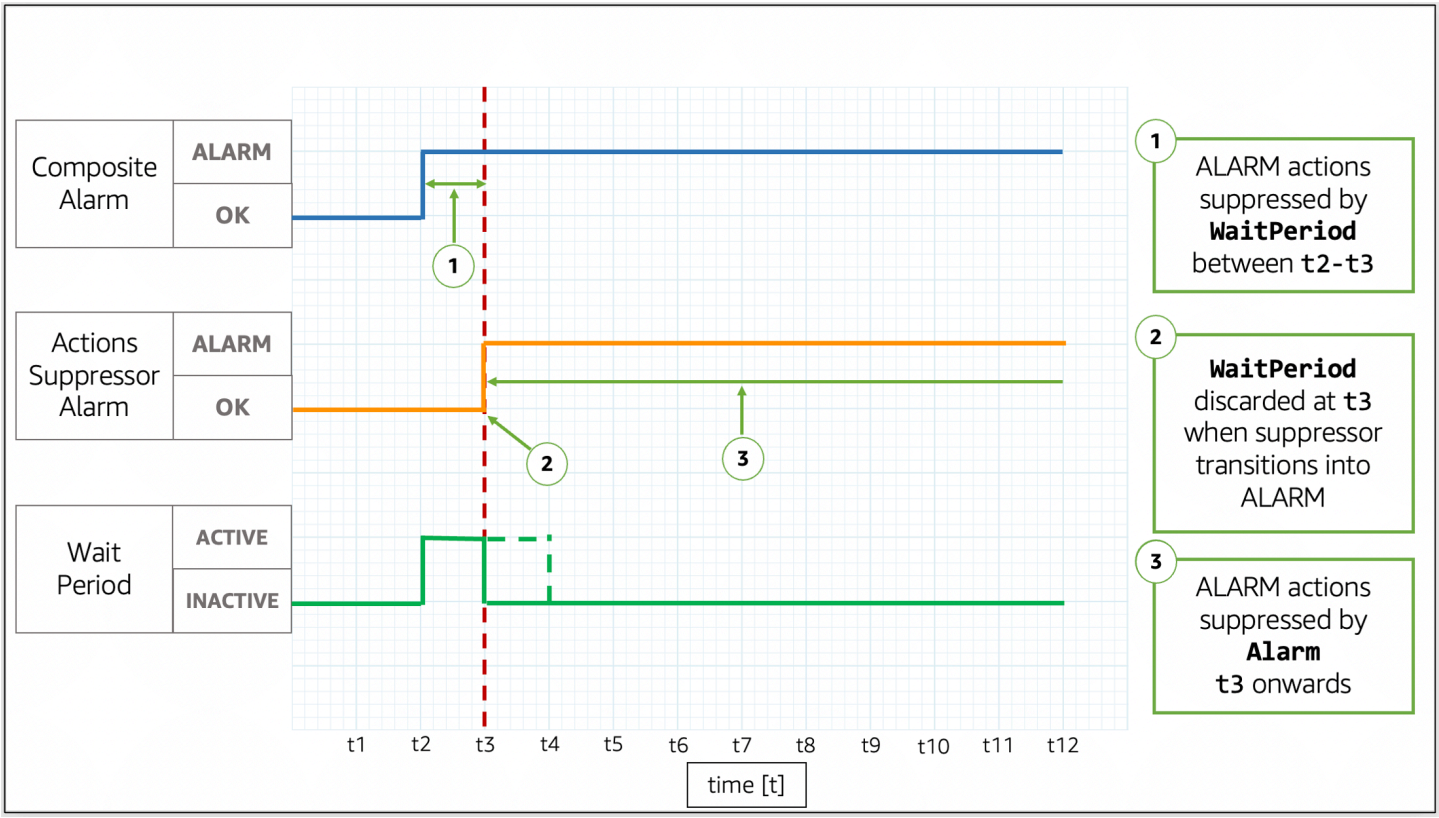
Beispiele

Beispiel 1: Aktionen werden nach **`WaitPeriod`** nicht unterdrückt



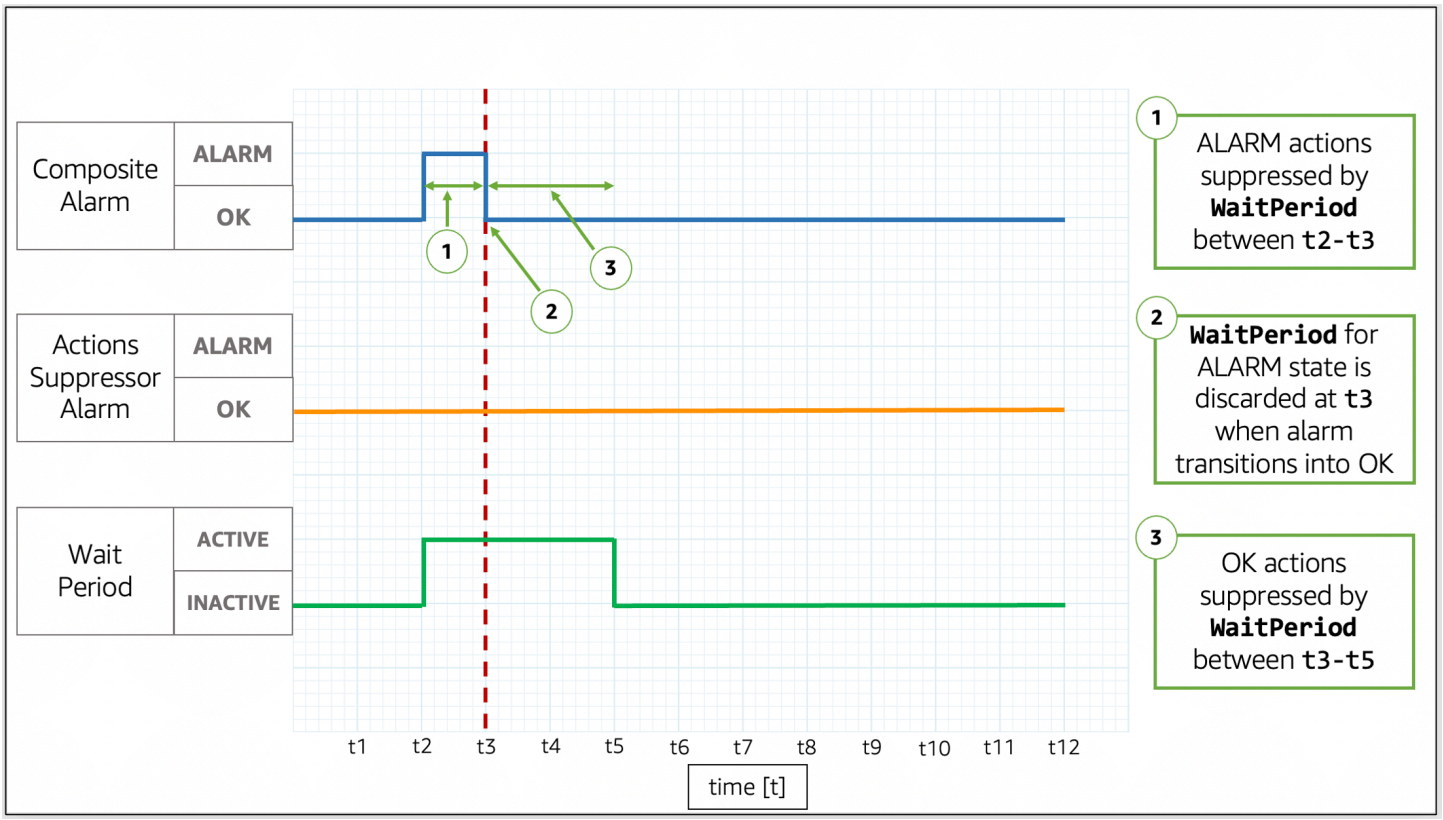
In der Abbildung ändert sich der Zustand des zusammengesetzten Alarms bei t2 von 0K in `ALARM`. Eine Wartezeit (`WaitPeriod`) beginnt bei t2 und endet bei t4, um zu verhindern, dass der zusammengesetzte Alarm Aktionen ausführt. Nach Ablauf der Wartezeit (`WaitPeriod`) bei t4 führt der zusammengesetzte Alarm seine Aktionen aus, da sich der Unterdrückungsalarm noch im Zustand 0K befindet.

Beispiel 2: Aktionen werden durch den Alarm unterdrückt, bevor **`WaitPeriod`** abläuft



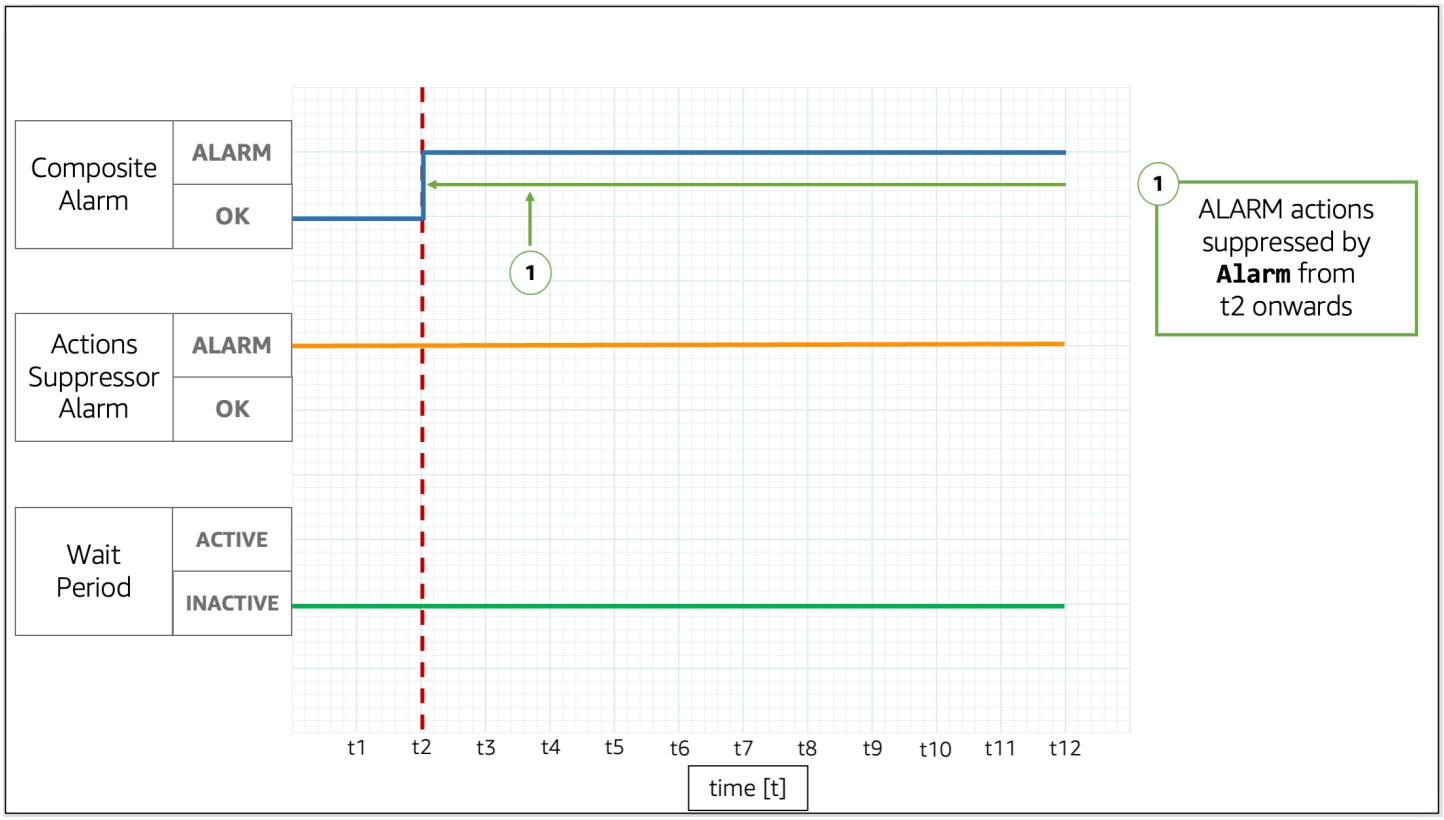
Im Image ändert sich der Zustand des zusammengesetzten Alarms bei t2 von `OK` in `ALARM`. Eine Wartezeit (`WaitPeriod`) beginnt bei t2 und endet bei t4. Dies gibt dem Unterdrückungsalarm Zeit, den Zustand bei t3 von `OK` in `ALARM` zu ändern. Da der Unterdrückungsalarm bei t3 den Zustand von `OK` in `ALARM` ändert, wird die bei t2 begonnene Wartezeit (`WaitPeriod`) verworfen, und der Unterdrückungsalarm verhindert nun, dass der zusammengesetzte Alarm Aktionen ausführt.

Beispiel 3: Zustandsübergang, wenn Aktionen durch **`WaitPeriod`** unterdrückt werden



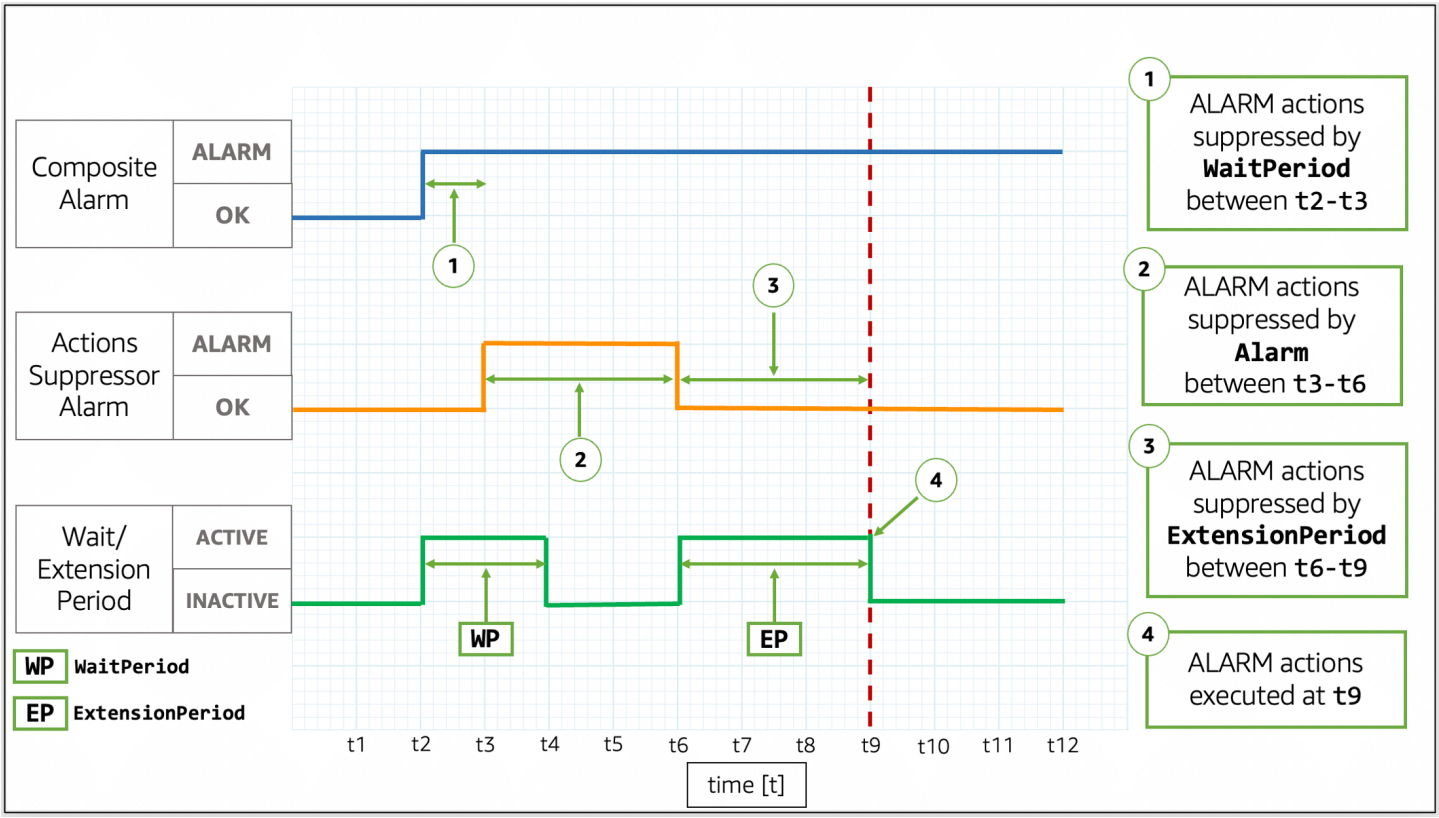
Im Image ändert sich der Zustand des zusammengesetzten Alarms bei t2 von 0K in ALARM. Eine Wartezeit (`WaitPeriod`) beginnt bei t2 und endet bei t4. Dies gibt dem Unterdrückungsalarm Zeit, den Zustand zu ändern. Der Zustand des zusammengesetzten Alarms ändert sich bei t3 wieder in 0K, wodurch die bei t2 begonnene Wartezeit (`WaitPeriod`) verworfen wird. Eine neue Wartezeit (`WaitPeriod`) beginnt bei t3 und endet bei t5. Nach Ablauf der neuen Wartezeit (`WaitPeriod`) bei t5 führt der zusammengesetzte Alarm seine Aktionen aus.

Beispiel 4: Zustandsübergang, wenn Aktionen durch einen Alarm unterdrückt werden



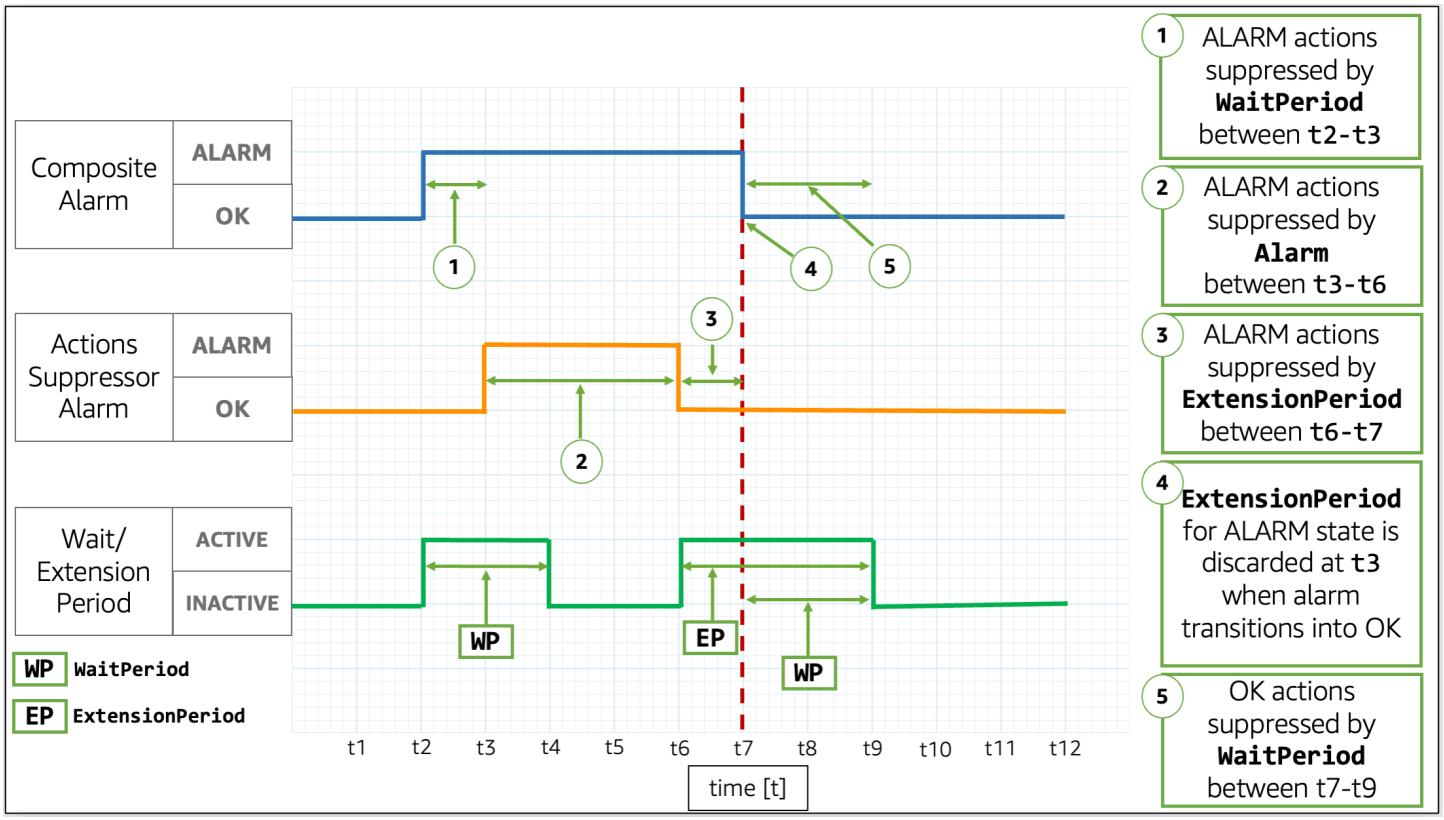
Im Image ändert sich der Zustand des zusammengesetzten Alarms bei t2 von `OK` in `ALARM`. Der Unterdrückungsalarm befindet sich bereits im Zustand `ALARM`. Der Unterdrückungsalarm verhindert, dass der zusammengesetzte Alarm Aktionen ausführt.

Beispiel 5: Aktionen werden nach **`ExtensionPeriod`** nicht unterdrückt



Im Image ändert sich der Zustand des zusammengesetzten Alarms bei t2 von OK in ALARM. Eine Wartezeit (`WaitPeriod`) beginnt bei t2 und endet bei t4. Dadurch hat der Unterdrückungsalarm Zeit, den Zustand bei t3 von OK in ALARM zu ändern, bevor die Aktionen des zusammengesetzten Alarms bis t6 unterdrückt werden. Da der Unterdrückungsalarm bei t3 den Zustand von OK in ALARM ändert, wird die bei t2 begonnene Wartezeit (`WaitPeriod`) verworfen. Bei t6 ändert sich der Zustand des Unterdrückungsalarms in OK. Ein Verlängerungszeitraum (`ExtensionPeriod`) beginnt bei t6 und endet bei t9. Nach Ablauf von `ExtensionPeriod` führt der zusammengesetzte Alarm seine Aktionen aus.

Beispiel 6: Zustandsübergang, wenn Aktionen durch **`ExtensionPeriod`** unterdrückt werden



Im Image ändert sich der Zustand des zusammengesetzten Alarms bei t2 von OK in ALARM. Eine Wartezeit (`WaitPeriod`) beginnt bei t2 und endet bei t4. Dadurch hat der Unterdrückungsalarm Zeit, den Zustand bei t3 von OK in ALARM zu ändern, bevor die Aktionen des zusammengesetzten Alarms bis t6 unterdrückt werden. Da der Unterdrückungsalarm bei t3 den Zustand von OK in ALARM ändert, wird die bei t2 begonnene Wartezeit (`WaitPeriod`) verworfen. Bei t6 ändert sich der Zustand des Unterdrückungsalarms wieder in OK. Ein Verlängerungszeitraum (`ExtensionPeriod`) beginnt bei t6 und endet bei t9. Wenn sich der Zustand des zusammengesetzten Alarm bei t7 wieder in OK ändert, wird der Verlängerungszeitraum (`ExtensionPeriod`) verworfen und eine neue Wartezeit (`WaitPeriod`) gestartet, die bei t7 beginnt und bei t9 endet.

> **Tip**
>
> Wenn Sie den Aktionsunterdrückungsalarm ersetzen, wird jegliche aktive Wartezeit (`WaitPeriod`) bzw. jeglicher aktiver Verlängerungszeitraum (`ExtensionPeriod`) verworfen.

Reagieren auf Alarmänderungen

CloudWatch kann Benutzer über zwei Arten von Alarmänderungen benachrichtigen: wenn sich der Status eines Alarms ändert und wenn die Konfiguration eines Alarms aktualisiert wird.

Wenn ein Alarm ausgewertet wird, kann er von einem Zustand in einen anderen wechseln, z. B. ALARM, OK oder INSUFFICIENT_DATA. Diese Alarmzustandsänderungen können einen möglichen Vorfall, eine Rückkehr zum Normalzustand oder die Nichtverfügbarkeit einer Metrik signalisieren. In solchen Fällen möchten Sie Benutzer möglicherweise mit einer der folgenden Optionen ansprechen oder diese benachrichtigen:

- Sie können den Alarm so konfigurieren, dass im Rahmen der Alarmaktionen eine Benachrichtigung an ein SNS-Thema gesendet wird. Ein SNS-Thema kann dann sowohl für application-to-application (A2A-) Nachrichten als auch für application-to-person (A2P-) Benachrichtigungen konfiguriert werden, einschließlich Kanälen wie E-Mail-Benachrichtigungen und SMS. Alle Ziele, die Sie für Ihr SNS-Thema definieren, erhalten die Alarmbenachrichtigung. Weitere Informationen finden Sie unter Amazon-SNS-Ereignisziele.
- Sie können Benachrichtigungen für Ereignisse zur Änderung des Alarmstatus konfigurieren. AWS Benutzerbenachrichtigungen bieten eine native Möglichkeit, solche Benachrichtigungen zu konfigurieren, und sind der empfohlene Ansatz.

CloudWatch Sendet außerdem Ereignisse an Amazon, EventBridge wenn sich der Status von Alarmen ändert und wenn Alarme erstellt, gelöscht oder aktualisiert werden. Sie können EventBridge Regeln schreiben, um Maßnahmen zu ergreifen, oder Sie können sich benachrichtigen lassen, EventBridge wenn Sie diese Ereignisse erhalten.

Themen

Benachrichtigen von Benutzern über Alarmänderungen

In diesem Abschnitt wird erklärt, wie Sie AWS Benutzerbenachrichtigungen oder Amazon Simple Notification Service verwenden können, um Benutzer über Alarmänderungen zu benachrichtigen.

AWS Benutzerbenachrichtigungen einrichten

Sie können AWS Benutzerbenachrichtigungen verwenden, um Zustellungskanäle einzurichten, über die Sie über Änderungen des CloudWatch Alarmstatus und über Änderungen der Konfiguration informiert werden. Sie erhalten eine Benachrichtigung, wenn ein Ereignis einer von Ihnen angegebenen Regel entspricht. Sie können Benachrichtigungen für Ereignisse über mehrere Kanäle erhalten, einschließlich E-Mail, AWS -Chatbot-Chat-Benachrichtigungen oder Push-Benachrichtigungen der mobilen AWS -Konsolenanwendung. Benachrichtigungen werden auch im Console Notifications Center angezeigt. Die Funktion für Benutzerbenachrichtigungen unterstützt eine Aggregation, wodurch die Anzahl der Benachrichtigungen, die Sie bei bestimmten Ereignissen erhalten, reduziert werden kann.

Die Benachrichtigungskonfigurationen, die Sie mit AWS Benutzerbenachrichtigungen erstellen, werden nicht auf die maximale Anzahl von Aktionen angerechnet, die Sie pro Zielalarmstatus konfigurieren können. Da AWS Benutzerbenachrichtigungen mit den an Amazon gesendeten Ereignissen übereinstimmen EventBridge, sendet Amazon Benachrichtigungen für alle Alarme in Ihrem Konto und ausgewählten Regionen, sofern Sie keinen erweiterten Filter angeben, um bestimmte Alarme oder Muster zuzulassen oder abzulehnen.

Das folgende Beispiel für einen erweiterten Filter entspricht einer Änderung des Alarmstatus von OK zu ALARM bei dem Alarm namens `ServerCpuTooHigh`.

```
{
"detail": {
    "alarmName": ["ServerCpuTooHigh"],
    "previousState": { "value": ["OK"] },
    "state": { "value": ["ALARM"] }
  }
}
```

Sie können alle Eigenschaften verwenden, die durch einen Alarm unter Ereignisse veröffentlicht wurden, um einen Filter zu erstellen. EventBridge Weitere Informationen finden Sie unter Alarmereignisse und EventBridge.

Einrichten von Amazon-SNS-Benachrichtigungen

Sie können Amazon Simple Notification Service verwenden, um sowohl application-to-application (A2A) als auch application-to-person (A2P) -Nachrichten zu senden, einschließlich mobiler Textnachrichten (SMS) und E-Mail-Nachrichten. Weitere Informationen finden Sie unter Amazon-SNS-Ereignisziele.

Für jeden Zustand, den ein Alarm annehmen kann, können Sie den Alarm so konfigurieren, dass er eine Nachricht an ein SNS-Thema sendet. Jedes Amazon-SNS-Thema, das Sie für einen Status eines bestimmten Alarms konfigurieren, zählt zur Begrenzung der Anzahl der Aktionen, die Sie für diesen Alarm und diesen Status konfigurieren können. Sie können Nachrichten an dasselbe Amazon-SNS-Thema von allen Alarmen in Ihrem Konto aus senden und dasselbe Amazon-SNS-Thema sowohl für Anwendungs- (A2A) als auch für Personenkunden (A2P) verwenden. Da diese Konfiguration auf Alarmebene vorgenommen wird, senden nur die von Ihnen konfigurierten Alarme Nachrichten an das ausgewählte Amazon-SNS-Thema.

Zunächst erstellen und abonnieren Sie ein Thema. Sie können optional eine Testnachricht für das Thema veröffentlichen. Ein Beispiel finden Sie unter Einrichten eines Amazon SNS SNS-Themas mit dem AWS Management Console. Weitere Informationen finden Sie auch unter Erste Schritte mit Amazon SNS.

Wenn Sie den AWS Management Console zur Erstellung Ihres CloudWatch Alarms verwenden möchten, können Sie dieses Verfahren alternativ überspringen, da Sie das Thema bei der Erstellung des Alarms erstellen können.

Wenn Sie einen CloudWatch Alarm erstellen, können Sie Aktionen für jeden Zielstatus hinzufügen, in den der Alarm eintritt. Fügen Sie eine Amazon-SNS-Benachrichtigung für den Status hinzu, über den Sie benachrichtigt werden möchten, und wählen Sie das Amazon-SNS-Thema, das Sie im vorherigen Schritt erstellt haben, um eine E-Mail-Benachrichtigung zu senden, wenn der Alarm in den ausgewählten Status eintritt.

> **Note**
>
> Wenn Sie ein Amazon SNS SNS-Thema erstellen, entscheiden Sie, ob Sie es zu einem Standardthema oder einem FIFO-Thema machen möchten. CloudWatch garantiert die Veröffentlichung aller Alarmmeldungen zu beiden Thementypen. Selbst wenn Sie ein FIFO-Thema verwenden, werden in seltenen Fällen die Benachrichtigungen CloudWatch an das Thema nicht in der richtigen Reihenfolge gesendet. Wenn Sie ein FIFO-Thema verwenden, legt der Alarm die Nachrichtengruppen-ID der Alarmbenachrichtigungen als Hash des ARN des Alarms fest.

Verhindern von verwirrten Stellvertreterproblemen

Wir empfehlen Ihnen, die Schlüssel `aws:SourceArn` und die `aws:SourceAccount` globalen Bedingungsschlüssel in der Amazon SNS-Ressourcenrichtlinie zu verwenden, die Ihnen den

Zugriff auf Ihre Amazon SNS-Ressourcen gewähren, damit Sie nicht auf Ihre Amazon SNS SNS-Ressourcen zugreifen können. CloudWatch

In der folgenden Beispiel-Ressourcenrichtlinie wird der `aws:SourceArn` Bedingungsschlüssel verwendet, um die `SNS:Publish` Zugriffsrechte auf CloudWatch Alarme im angegebenen Konto einzuschränken.

```
{
    "Statement": [{
        "Effect": "Allow",
        "Principal": {
            "Service": "cloudwatch.amazonaws.com"
        },
        "Action": "SNS:Publish",
        "Resource": "arn:aws:sns:us-east-2:444455556666:MyTopic",
        "Condition": {
            "ArnLike": {
                "aws:SourceArn": "arn:aws:cloudwatch:us-east-2:111122223333:alarm:*"
            },
            "StringEquals": {
                "aws:SourceAccount": "111122223333"
            }
        }
    }]
}
```

Wenn ein Alarm-ARN irgendwelche Nicht-ASCII-Zeichen enthält, verwenden Sie nur den globalen Bedingungsschlüssel `aws:SourceAccount` zur Begrenzung der Berechtigungen.

Einrichten eines Amazon SNS SNS-Themas mit dem AWS Management Console

Zunächst erstellen und abonnieren Sie ein Thema. Sie können optional eine Testnachricht für das Thema veröffentlichen.

So erstellen Sie ein SNS-Thema

1. Öffnen Sie die Amazon SNS-Konsole unter https://console.aws.amazon.com/sns/v3/home.
2. Wählen Sie auf dem Amazon-SNS-Dashboard unter Common actions die Option Create Topic.
3. Geben Sie im Dialogfeld Create new topic (Neues Thema erstellen) unter Topic name (Themaname) einen Namen für das Thema ein (z. B. **`my-topic`**).
4. Wählen Sie Thema erstellen aus.

5. Kopieren Sie den Topic ARN (Themen-ARN) für die nächste Aufgabe (z. B. arn:aws:sns:us-east-1:111122223333:my-topic).

So abonnieren Sie ein SNS-Thema

1. Öffnen Sie die Amazon SNS-Konsole unter https://console.aws.amazon.com/sns/v3/home.
2. Wählen Sie im Navigationsbereich Abonnements und Abonnement erstellen aus.
3. Fügen Sie im Dialogfeld für Create subscription für Topic ARN den ARN des Themas ein, den Sie im vorherigen Schritt erstellt haben.
4. Wählen Sie unter Protocol (Protokoll) die Option Email (E-Mail) aus.
5. Geben Sie für Endpoint (Endpunkt) eine E-Mail-Adresse ein, mit der Sie die Benachrichtigung erhalten können, und wählen Sie dann Create subscription (Abonnement erstellen).
6. Öffnen Sie in Ihrer E-Mail-Anwendung die Nachricht unter AWS Benachrichtigungen und bestätigen Sie Ihr Abonnement.

   Ihr Webbrowser zeigt eine Bestätigungsantwort vom Amazon SNS an.

So veröffentlichen Sie eine Testnachricht in einem SNS-Thema

1. Öffnen Sie die Amazon SNS-Konsole unter https://console.aws.amazon.com/sns/v3/home.
2. Wählen Sie im Navigationsbereich Themen aus.
3. Wählen Sie auf der Seite Topics ein Thema und dann Publish to topic aus.
4. Geben Sie auf der Seite Publish a message (Veröffentlichen einer Nachricht) unter Subject (Betreff) eine Betreffzeile für Ihre Nachricht und unter Message (Nachricht) eine kurze Nachricht ein.
5. Wählen Sie Publish Message aus.
6. Überprüfen Sie Ihren E-Mail-Eingang, um zu bestätigen, dass Sie die Nachricht erhalten haben.

Einrichten eines SNS-Themas mit dem AWS CLI

Zuerst erstellen Sie ein SNS-Thema und veröffentlichen dann eine Nachricht direkt an das Thema, um zu testen, ob Sie es richtig konfiguriert haben.

So richten Sie ein SNS-Thema ein

1. Erstellen Sie das Thema mit dem Befehl create-topic wie folgt.

```
aws sns create-topic --name my-topic
```

Amazon SNS gibt einen ARN für das Thema im folgenden Format zurück:

```
{
    "TopicArn": "arn:aws:sns:us-east-1:111122223333:my-topic"
}
```

2. Abonnieren Sie die E-Mail-Adresse für das Thema mit dem Befehl subscribe. Wenn die Abonnementanfrage erfolgreich ist, erhalten Sie eine Bestätigungs-E-Mail.

```
aws sns subscribe --topic-arn arn:aws:sns:us-east-1:111122223333:my-topic --
protocol email --notification-endpoint my-email-address
```

Amazon SNS gibt Folgendes zurück:

```
{
    "SubscriptionArn": "pending confirmation"
}
```

3. Öffnen Sie in Ihrer E-Mail-Anwendung die Nachricht unter AWS Benachrichtigungen und bestätigen Sie Ihr Abonnement.

Ihr Webbrowser zeigt eine Bestätigungsantwort vom Amazon Simple Notification Service an.

4. Überprüfen Sie das Abonnement mit dem list-subscriptions-by-topicBefehl.

```
aws sns list-subscriptions-by-topic --topic-arn arn:aws:sns:us-
east-1:111122223333:my-topic
```

Amazon SNS gibt Folgendes zurück:

```
{
  "Subscriptions": [
    {
        "Owner": "111122223333",
        "Endpoint": "me@mycompany.com",
        "Protocol": "email",
        "TopicArn": "arn:aws:sns:us-east-1:111122223333:my-topic",
```

```
        "SubscriptionArn": "arn:aws:sns:us-east-1:111122223333:my-topic:64886986-
bf10-48fb-a2f1-dab033aa67a3"
    }
  ]
}
```

5. (Optional) Veröffentlichen Sie mit dem Befehl publish eine Testnachricht für das Thema.

```
aws sns publish --message "Verification" --topic arn:aws:sns:us-
east-1:111122223333:my-topic
```

Amazon SNS gibt Folgendes zurück.

```
{
    "MessageId": "42f189a0-3094-5cf6-8fd7-c2dde61a4d7d"
}
```

6. Überprüfen Sie Ihren E-Mail-Eingang, um zu bestätigen, dass Sie die Nachricht erhalten haben.

Alarmereignisse und EventBridge

CloudWatch sendet Ereignisse an Amazon, EventBridge wenn ein CloudWatch Alarm erstellt, aktualisiert, gelöscht oder der Alarmstatus geändert wird. Sie können diese Ereignisse verwenden EventBridge, um Regeln zu schreiben, die Maßnahmen ergreifen, z. B. Sie benachrichtigen, wenn sich der Status eines Alarms ändert. Weitere Informationen finden Sie unter Was ist Amazon EventBridge?

CloudWatch garantiert die Übermittlung von Ereignissen zur Änderung des Alarmstatus an EventBridge.

Beispielereignisse von CloudWatch

Dieser Abschnitt enthält Beispielereignisse von CloudWatch.

Zustandsänderung für einen Einzelmetrik-Alarm

```
{
    "version": "0",
    "id": "c4c1c1c9-6542-e61b-6ef0-8c4d36933a92",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
```

```
    "account": "123456789012",
    "time": "2019-10-02T17:04:40Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServerCpuTooHigh"
    ],
    "detail": {
        "alarmName": "ServerCpuTooHigh",
        "configuration": {
            "description": "Goes into alarm when server CPU utilization is too high!",
            "metrics": [
                {
                    "id": "30b6c6b2-a864-43a2-4877-c09a1afc3b87",
                    "metricStat": {
                        "metric": {
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            },
                            "name": "CPUUtilization",
                            "namespace": "AWS/EC2"
                        },
                        "period": 300,
                        "stat": "Average"
                    },
                    "returnData": true
                }
            ]
        },
        "previousState": {
            "reason": "Threshold Crossed: 1 out of the last 1 datapoints
 [0.0666851903306472 (01/10/19 13:46:00)] was not greater than the threshold (50.0)
 (minimum 1 datapoint for ALARM -> OK transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-01T13:56:40.985+0000\",\"startDate\":\"2019-10-01T13:46:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[0.0666851903306472],
\"threshold\":50.0}",
            "timestamp": "2019-10-01T13:56:40.987+0000",
            "value": "OK"
        },
        "state": {
            "reason": "Threshold Crossed: 1 out of the last 1 datapoints
 [99.50160229693434 (02/10/19 16:59:00)] was greater than the threshold (50.0) (minimum
 1 datapoint for OK -> ALARM transition).",
```

```
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-02T17:04:40.985+0000\",\"startDate\":\"2019-10-02T16:59:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[99.50160229693434],
\"threshold\":50.0}",
            "timestamp": "2019-10-02T17:04:40.989+0000",
            "value": "ALARM"
        }
    }
}
```

Zustandsänderung für einen Metrikmathematik-Alarm

```
{
    "version": "0",
    "id": "2dde0eb1-528b-d2d5-9ca6-6d590caf2329",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2019-10-02T17:20:48Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:TotalNetworkTrafficTooHigh"
    ],
    "detail": {
        "alarmName": "TotalNetworkTrafficTooHigh",
        "configuration": {
            "description": "Goes into alarm if total network traffic exceeds 10Kb",
            "metrics": [
                {
                    "expression": "SUM(METRICS())",
                    "id": "e1",
                    "label": "Total Network Traffic",
                    "returnData": true
                },
                {
                    "id": "m1",
                    "metricStat": {
                        "metric": {
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            },
                            "name": "NetworkIn",
                            "namespace": "AWS/EC2"
```

```
                        },
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                },
                {
                    "id": "m2",
                    "metricStat": {
                        "metric": {
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            },
                            "name": "NetworkOut",
                            "namespace": "AWS/EC2"
                        },
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                }
            ]
        },
        "previousState": {
            "reason": "Unchecked: Initial alarm creation",
            "timestamp": "2019-10-02T17:20:03.642+0000",
            "value": "INSUFFICIENT_DATA"
        },
        "state": {
            "reason": "Threshold Crossed: 1 out of the last 1 datapoints [45628.0
 (02/10/19 17:10:00)] was greater than the threshold (10000.0) (minimum 1 datapoint for
 OK -> ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-02T17:20:48.551+0000\",\"startDate\":\"2019-10-02T17:10:00.000+0000\",
\"period\":300,\"recentDatapoints\":[45628.0],\"threshold\":10000.0}",
            "timestamp": "2019-10-02T17:20:48.554+0000",
            "value": "ALARM"
        }
    }
}
```

Zustandsänderung für einen Anomalieerkennungs-Alarm

```
{
    "version": "0",
    "id": "daafc9f1-bddd-c6c9-83af-74971fcfc4ef",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2019-10-03T16:00:04Z",
    "region": "us-east-1",
    "resources": ["arn:aws:cloudwatch:us-east-1:123456789012:alarm:EC2 CPU Utilization
 Anomaly"],
    "detail": {
        "alarmName": "EC2 CPU Utilization Anomaly",
        "state": {
            "value": "ALARM",
            "reason": "Thresholds Crossed: 1 out of the last 1 datapoints [0.0
 (03/10/19 15:58:00)] was less than the lower thresholds [0.020599444741798756] or
 greater than the upper thresholds [0.3006915352732461] (minimum 1 datapoint for OK ->
 ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-03T16:00:04.650+0000\",\"startDate\":\"2019-10-03T15:58:00.000+0000\",
\"period\":60,\"recentDatapoints\":[0.0],\"recentLowerThresholds\":
[0.020599444741798756],\"recentUpperThresholds\":[0.3006915352732461]}",
            "timestamp": "2019-10-03T16:00:04.653+0000"
        },
        "previousState": {
            "value": "OK",
            "reason": "Thresholds Crossed: 1 out of the last 1 datapoints
 [0.166666666664241 (03/10/19 15:57:00)] was not less than the lower thresholds
 [0.0206719426210418] or not greater than the upper thresholds [0.30076870222143803]
 (minimum 1 datapoint for ALARM -> OK transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-03T15:59:04.670+0000\",\"startDate\":\"2019-10-03T15:57:00.000+0000\",
\"period\":60,\"recentDatapoints\":[0.166666666664241],\"recentLowerThresholds\":
[0.0206719426210418],\"recentUpperThresholds\":[0.30076870222143803]}",
            "timestamp": "2019-10-03T15:59:04.672+0000"
        },
        "configuration": {
            "description": "Goes into alarm if CPU Utilization is out of band",
            "metrics": [{
                "id": "m1",
                "metricStat": {
                    "metric": {
                        "namespace": "AWS/EC2",
```

```
                        "name": "CPUUtilization",
                        "dimensions": {
                            "InstanceId": "i-12345678901234567"
                        }
                    },
                    "period": 60,
                    "stat": "Average"
                },
                "returnData": true
            }, {
                "id": "ad1",
                "expression": "ANOMALY_DETECTION_BAND(m1, 0.8)",
                "label": "CPUUtilization (expected)",
                "returnData": true
            }]
        }
    }
}
```

Zustandsänderung für einen zusammengesetzten Alarm mit einem Unterdrückungsalarm

```
{
    "version": "0",
    "id": "d3dfc86d-384d-24c8-0345-9f7986db0b80",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-22T15:57:45Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "state": {
            "actionsSuppressedBy": "WaitPeriod",
            "actionsSuppressedReason": "Actions suppressed by WaitPeriod",
            "value": "ALARM",
            "reason": "arn:aws:cloudwatch:us-
east-1:123456789012:alarm:SuppressionDemo.EventBridge.FirstChild transitioned to ALARM
 at Friday 22 July, 2022 15:57:45 UTC",
```

```
            "reasonData": "{\"triggeringAlarms\":[{\"arn\":\"arn:aws:cloudwatch:us-
east-1:123456789012:alarm:ServerCpuTooHigh\",\"state\":{\"value\":\"ALARM\",\"timestamp
\":\"2022-07-22T15:57:45.394+0000\"}}]}",
            "timestamp": "2022-07-22T15:57:45.394+0000"
        },
        "previousState": {
            "value": "OK",
            "reason": "arn:aws:cloudwatch:us-
east-1:123456789012:alarm:SuppressionDemo.EventBridge.Main was created and its alarm
 rule evaluates to OK",
            "reasonData": "{\"triggeringAlarms\":[{\"arn\":\"arn:aws:cloudwatch:us-
east-1:123456789012:alarm:TotalNetworkTrafficTooHigh\",\"state\":{\"value\":\"OK\",
\"timestamp\":\"2022-07-14T16:28:57.770+0000\"}},{\"arn\":\"arn:aws:cloudwatch:us-
east-1:123456789012:alarm:ServerCpuTooHigh\",\"state\":{\"value\":\"OK\",\"timestamp\":
\"2022-07-14T16:28:54.191+0000\"}}]}",
            "timestamp": "2022-07-22T15:56:14.552+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 180
        }
    }
}
```

Erstellen eines zusammengesetzten Alarms

```
{

    "version": "0",
    "id": "91535fdd-1e9c-849d-624b-9a9f2b1d09d0",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-03-03T17:06:22Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
```

```
        "operation": "create",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-03-03T17:06:22.289+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "alarmName": "ServiceAggregatedAlarm",
            "description": "Aggregated monitor for instance",
            "actionsEnabled": true,
            "timestamp": "2022-03-03T17:06:22.289+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Erstellung eines zusammengesetzten Alarms mit einem Unterdrückungsalarm

```
{
    "version": "0",
    "id": "454773e1-09f7-945b-aa2c-590af1c3f8e0",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-14T13:59:46Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "operation": "create",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-07-14T13:59:46.425+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
```

```
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 180,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:46.425+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Aktualisierung eines Metrik-Alarms

```
{

    "version": "0",
    "id": "bc7d3391-47f8-ae47-f457-1b4d06118d50",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-03-03T17:06:34Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServerCpuTooHigh"
    ],
    "detail": {
        "alarmName": "ServerCpuTooHigh",
        "operation": "update",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-03-03T17:06:13.757+0000"
        },
        "configuration": {
            "evaluationPeriods": 1,
            "threshold": 80,
            "comparisonOperator": "GreaterThanThreshold",
            "treatMissingData": "ignore",
            "metrics": [
                {
                    "id": "86bfa85f-b14c-ebf7-8916-7da014ce23c0",
                    "metricStat": {
                        "metric": {
```

```
                        "namespace": "AWS/EC2",
                        "name": "CPUUtilization",
                        "dimensions": {
                            "InstanceId": "i-12345678901234567"
                        }
                    },
                    "period": 300,
                    "stat": "Average"
                },
                "returnData": true
            }
        ],
        "alarmName": "ServerCpuTooHigh",
        "description": "Goes into alarm when server CPU utilization is too high!",
        "actionsEnabled": true,
        "timestamp": "2022-03-03T17:06:34.267+0000",
        "okActions": [],
        "alarmActions": [],
        "insufficientDataActions": []
    },
    "previousConfiguration": {
        "evaluationPeriods": 1,
        "threshold": 70,
        "comparisonOperator": "GreaterThanThreshold",
        "treatMissingData": "ignore",
        "metrics": [
            {
                "id": "d6bfa85f-893e-b052-a58b-4f9295c9111a",
                "metricStat": {
                    "metric": {
                        "namespace": "AWS/EC2",
                        "name": "CPUUtilization",
                        "dimensions": {
                            "InstanceId": "i-12345678901234567"
                        }
                    },
                    "period": 300,
                    "stat": "Average"
                },
                "returnData": true
            }
        ],
        "alarmName": "ServerCpuTooHigh",
        "description": "Goes into alarm when server CPU utilization is too high!",
```

```
            "actionsEnabled": true,
            "timestamp": "2022-03-03T17:06:13.757+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Aktualisierung eines zusammengesetzten Alarms mit einem Unterdrückungsalarm

```
{
    "version": "0",
    "id": "4c6f4177-6bd5-c0ca-9f05-b4151c54568b",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-14T13:59:56Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "operation": "update",
        "state": {
            "actionsSuppressedBy": "WaitPeriod",
            "value": "ALARM",
            "timestamp": "2022-07-14T13:59:46.425+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 360,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:56.290+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        },
```

```
        "previousConfiguration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 180,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:46.425+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Löschen eines mathematischen Metrik-Alarms

```
{

    "version": "0",
    "id": "f171d220-9e1c-c252-5042-2677347a83ed",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-03-03T17:07:13Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:TotalNetworkTrafficTooHigh"
    ],
    "detail": {
        "alarmName": "TotalNetworkTrafficTooHigh",
        "operation": "delete",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-03-03T17:06:17.672+0000"
        },
        "configuration": {
            "evaluationPeriods": 1,
            "threshold": 10000,
            "comparisonOperator": "GreaterThanThreshold",
            "treatMissingData": "ignore",
            "metrics": [{
```

```
                    "id": "m1",
                    "metricStat": {
                        "metric": {
                            "namespace": "AWS/EC2",
                            "name": "NetworkIn",
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            }
                        },
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                },
                {
                    "id": "m2",
                    "metricStat": {
                        "metric": {
                            "namespace": "AWS/EC2",
                            "name": "NetworkOut",
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            }
                        },
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                },
                {
                    "id": "e1",
                    "expression": "SUM(METRICS())",
                    "label": "Total Network Traffic",
                    "returnData": true
                }
            ],
            "alarmName": "TotalNetworkTrafficTooHigh",
            "description": "Goes into alarm if total network traffic exceeds 10Kb",
            "actionsEnabled": true,
            "timestamp": "2022-03-03T17:06:17.672+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
```

```
        }
    }
}
```

Löschung eines zusammengesetzten Alarms mit einem Unterdrückungsalarm

```
{
    "version": "0",
    "id": "e34592a1-46c0-b316-f614-1b17a87be9dc",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-14T14:00:01Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "operation": "delete",
        "state": {
            "actionsSuppressedBy": "WaitPeriod",
            "value": "ALARM",
            "timestamp": "2022-07-14T13:59:46.425+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 360,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:56.290+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Verwalten von Alarmen

Bearbeiten oder löschen Sie einen CloudWatch Alarm

Sie können einen vorhandenen Alarm bearbeiten oder löschen.

Sie können den Namen eines vorhandenen Alarms nicht ändern. Sie können einen Alarm kopieren und dem neuen Alarm einen anderen Namen geben. Um einen Alarm zu kopieren, aktivieren Sie das Kontrollkästchen neben dem Alarmnamen in der Alarmliste und wählen Sie Action (Aktion), Copy (Kopieren).

So bearbeiten Sie einen Alarm

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich zuerst Alarme und dann Alle Alarme aus.
3. Wählen Sie den Namen des Alarms.
4. Um Tags hinzuzufügen oder zu entfernen, wählen Sie die Registerkarte Tags und dann Tags verwalten.
5. Um andere Teile des Alarms zu bearbeiten, wählen Sie Aktionen, Bearbeiten.

   Die Seite Specify metric and conditions (Metrik und Bedingungen festlegen) wird angezeigt, die ein Diagramm und andere Informationen über die von Ihnen ausgewählte Metrik und Statistik anzeigt.
6. Um die Metrik zu ändern, wählen Sie Edit (Bearbeiten), wählen Sie die Registerkarte All metrics (Alle Metriken) und führen Sie eine der folgenden Aktionen aus:
   - Wählen Sie den Service-Namenspace, der die gewünschte Metrik enthält. Fahren Sie mit der Auswahl der sichtbar werdenden Optionen fort, um die Auswahl einzugrenzen. Wenn eine Liste von Metriken angezeigt wird, aktivieren Sie das Kontrollkästchen neben der gewünschten Metrik.
   - Geben Sie im Suchfeld den Namen einer Metrik, Dimension oder Ressourcen-ID ein und drücken Sie die Eingabetaste. Wählen Sie dann eines der Ergebnisse aus und fahren Sie fort, bis eine Liste der Metriken erscheint. Aktivieren Sie das Kontrollkästchen neben der gewünschten Metrik.

   Wählen Sie Select metric (Metrik auswählen) aus.
7. Um andere Aspekte des Alarms zu ändern, wählen Sie die entsprechenden Optionen. Um zu ändern, wie viele Datenpunkte überschritten werden müssen, damit der Alarm in den Status

ALARM wechselt, oder um zu ändern, wie fehlende Daten behandelt werden, wählen Sie Additional configuration (Zusätzliche Konfiguration).

8. Wählen Sie Weiter aus.
9. Unter Notification (Benachrichtigung), Auto Scaling (Automatische Skalierungsaktion) und EC2 action (EC2-Aktion) können Sie optional die Aktionen bearbeiten, die bei Auslösung des Alarms durchgeführt werden. Wählen Sie anschließend Weiter.
10. Optional kann die Alarmbeschreibung geändert werden.

    Sie können den Namen eines vorhandenen Alarms nicht ändern. Sie können einen Alarm kopieren und dem neuen Alarm einen anderen Namen geben. Um einen Alarm zu kopieren, aktivieren Sie das Kontrollkästchen neben dem Alarmnamen in der Alarmliste und wählen Sie Action (Aktion), Copy (Kopieren).
11. Wählen Sie Weiter aus.
12. Bestätigen Sie unter Preview and create (Vorschau und erstellen), dass die Informationen und Bedingungen den Anforderungen entsprechen, und wählen Sie dann Update alarm (Alarm aktualisieren).

So aktualisieren Sie eine E-Mail-Benachrichtigungsliste, die über die Amazon-SNS-Konsole erstellt wurde

1. Öffnen Sie die Amazon SNS-Konsole unter https://console.aws.amazon.com/sns/v3/home.
2. Wählen Sie im Navigationsbereich Topics (Themen) und wählen Sie dann den ARN für Ihre Benachrichtigungsliste (Thema).
3. Führen Sie eine der folgenden Aktionen aus:

   - Zum Hinzufügen einer E-Mail-Adresse wählen Sie Create subscription aus. Wählen Sie unter Protocol (Protokoll) die Option Email (E-Mail) aus. Geben Sie unter Endpoint (Endpunkt) die E-Mail-Adresse des neuen Empfängers ein. Wählen Sie Create subscription (Abonnement erstellen) aus.
   - Zum Entfernen einer E-Mail-Adresse wählen Sie Subscription ID. Wählen Sie Other subscription actions, Delete subscriptions aus.
4. Wählen Sie Publish to topic (An Thema veröffentlichen).

So löschen Sie einen Alarm

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Klicken Sie im Navigationsbereich auf Alarms (Alarme).
3. Aktivieren Sie das Kontrollkästchen links neben dem Namen des Alarms, und wählen Sie Aktionen, Löschen.
4. Wählen Sie Löschen aus.

Auto Scaling Scaling-Alarme ausblenden

Wenn Sie Ihre Alarme im anzeigen, können Sie die Alarme ausblenden AWS Management Console, die sich sowohl auf Amazon EC2 Auto Scaling als auch auf Application Auto Scaling beziehen. Dieses Feature steht nur in der AWS Management Console zur Verfügung.

So blenden Sie Auto-Scaling-Alarme vorübergehend aus

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Klicken Sie im Navigationsbereich auf Alarme, dann auf Alle Alarme und schließlich auf Auto Scaling-Alarme ausblenden.

Anwendungsfälle und Beispiele für Alarme

In den folgenden Abschnitten finden Sie Beispiele und Anleitungen für Alarme für häufige Anwendungsfälle.

Erstellen Sie einen Abrechnungsalarm, um Ihre geschätzten AWS Gebühren zu überwachen

Sie können Ihre geschätzten AWS Gebühren mithilfe von Amazon überwachen CloudWatch. Wenn Sie die Überwachung der geschätzten Gebühren für Ihr AWS Konto aktivieren, werden die geschätzten Gebühren berechnet und mehrmals täglich CloudWatch als Metrikdaten an sie gesendet.

Die metrischen Fakturierungsdaten werden in der USA Ost (Nord-Virginia)-Region gespeichert und stellen die weltweiten Gebühren dar. Diese Daten beinhalten die geschätzten Gebühren für jeden Dienst AWS , den Sie nutzen, sowie die geschätzte Gesamtsumme Ihrer AWS Gebühren.

Der Alarm wird ausgelöst, wenn Ihre Kontoabrechnung den von Ihnen festgelegten Grenzwert überschreitet. Er wird nur ausgelöst, wenn die aktuelle Abrechnung den Grenzwert überschreitet. Der Service werden keine Prognosen auf der Grundlage Ihrer bisherigen Nutzung im Monat verwendet.

Wenn Sie einen Abrechnungsalarm zu einem Zeitpunkt erstellen, an dem Ihre Gebühren den Grenzwert bereits überschritten haben, wechselt der Alarm sofort in den `ALARM`-Status.

> **Note**
>
> Informationen zur Analyse von CloudWatch Gebühren, die Ihnen bereits in Rechnung gestellt wurden, finden Sie unterCloudWatch Abrechnung und Kosten.

Aufgaben

- Aktivieren von Abrechnung-Alarmen
- Erstellen einer Fakturierungsbenachrichtigung
- Löschen eines Abrechnung-Alarms

Aktivieren von Abrechnung-Alarmen

Bevor Sie einen Alarm für Ihre geschätzten Gebühren erstellen können, müssen Sie die Fakturierungsbenachrichtigungen aktivieren, damit Sie Ihre geschätzten AWS Gebühren überwachen und anhand von Abrechnungskennzahlen einen Alarm erstellen können. Nachdem Sie die Abrechnung-Alarme aktiviert haben, können Sie die Datenerfassung nicht deaktivieren, aber Sie können alle von Ihnen erstellten Abrechnung-Alarme löschen.

Wenn Sie Gebührenlimit-Warnungen zum ersten Mal aktivieren, dauert es etwa 15 Minuten, bevor Sie die Gebührendaten anzeigen und Gebührenlimit-Warnungen einrichten können.

Voraussetzungen

- Sie müssen mit den Anmeldeinformationen des Stammbenutzers des Kontos oder als IAM-Benutzer angemeldet sein, dem die Berechtigung zum Anzeigen von Fakturierungsinformationen erteilt wurde.
- Für konsolidierte Fakturierungskonten können die Gebührendaten für die verknüpften Konten durch eine Anmeldung als Zahlungskonto abgerufen werden. Sie können die Gebührendaten für die geschätzten Gesamtkosten und die geschätzten Gebühren nach Service für die einzelnen verknüpften Konten sowie für das konsolidierte Konto anzeigen.

- In einem konsolidierten Fakturierungskonto werden die Metriken für verknüpfte Konten nur erfasst, wenn das Zahlerkonto die Voreinstellung Fakturierungsbenachrichtigungen erhalten aktiviert. Wenn Sie ändern, welches Konto Ihr Management-/Zahler-Konto ist, müssen Sie die Gebührenlimit-Warnung im neuen Management-/Zahler-Konto aktivieren.
- Das Konto darf nicht Teil des Amazon Partner Network (APN) sein, da Abrechnungskennzahlen nicht CloudWatch für APN-Konten veröffentlicht werden. Weitere Informationen finden Sie unter AWS -Partnernetzwerk.

So aktivieren Sie die Überwachung der geschätzten Gebühren

1. Öffnen Sie die AWS Billing Konsole unter https://console.aws.amazon.com/billing/.
2. Wählen Sie im Navigationsbereich die Option Fakturierungseinstellungen aus.
3. Wählen Sie unter Präferenzen für Warnungen die Option Bearbeiten aus.
4. Wählen Sie „ CloudWatch Rechnungsbenachrichtigungen empfangen“.
5. Klicken Sie auf Präferenzen speichern.

Erstellen einer Fakturierungsbenachrichtigung

> **Important**
>
> Bevor Sie einen Fakturierungsalarm erstellen, müssen Sie Ihre Region auf USA Ost (Nord-Virginia) setzen. Die metrischen Fakturierungsdaten werden in dieser Region gespeichert und stellen die weltweiten Gebühren dar. Sie müssen Gebührenlimit-Warnungen in Ihrem Konto oder im Management-/Zahler-Konto aktivieren (wenn Sie die konsolidierte Abrechnung verwenden). Weitere Informationen finden Sie unter Aktivieren von Abrechnung-Alarmen.

In diesem Verfahren erstellen Sie einen Alarm, der eine Benachrichtigung sendet, wenn Ihre geschätzten Gebühren einen definierten Schwellenwert AWS überschreiten.

So erstellen Sie mit der CloudWatch Konsole einen Abrechnungsalarm

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Alarme und dann Alle Alarme aus.
3. Wählen Sie Create alarm (Alarm erstellen) aus.

4. Wählen Sie Select metric (Metrik auswählen) aus. Wählen Sie unter Browse (Durchsuchen) die Option Billing (Fakturierung) und dann Estimated Charge (Geschätzte Gesamtgebühr) aus.

   > **Note**
   >
   > Wenn die Metrik Fakturierung/Geschätzte Gesamtgebühr nicht angezeigt wird, aktivieren Sie Fakturierungsalarme und ändern Sie Ihre Region in USA Ost (Nord-Virginia). Weitere Informationen finden Sie unter Aktivieren von Abrechnung-Alarmen.

5. Wählen Sie das Feld für die EstimatedChargesMetrik aus, und wählen Sie dann Metrik auswählen aus.
6. Wählen Sie für Statistic (Statistik) Maximum aus.
7. Wählen Sie als Period (Zeitraum) 6 hours (6 Stunden) aus.
8. Wählen Sie für Threshold type (Schwellenwerttyp) die Option Static (Statisch) aus.
9. Für Wann immer EstimatedCharges es ist. , wählen Sie Größer.
10. Unter als . . . , definieren Sie den Wert, bei dem Ihr Alarm ausgelöst werden soll. Zum Beispiel, **`200`** USD.

    Die EstimatedChargesmetrischen Werte sind nur in US-Dollar (USD) angegeben, und die Währungsumrechnung wird von Amazon Services LLC bereitgestellt. Weitere Informationen finden Sie unter Was ist AWS Billing? .

    > **Note**
    >
    > Nachdem Sie einen Schwellenwert definiert haben, werden Ihre geschätzten Gebühren für den aktuellen Monat im Vorschaudiagramm angezeigt.

11. Wählen Sie Zusätzliche Konfiguration und führen Sie Folgendes aus:

    - Geben Sie für Datapoints to alarm (zu alarmierende Datenpunkte) 1 out of 1 (1 von 1) an.
    - Wählen Sie für Missing data treatment (Behandlung fehlender Daten) die Option Treat missing data as missing (Fehlende Daten als fehlend behandeln) aus.

12. Wählen Sie Weiter aus.
13. Stellen Sie sicher, dass unter Benachrichtigung die Option Bei Alarm ausgewählt ist. Legen Sie dann ein Amazon-SNS-Thema fest, das benachrichtigt werden soll, wenn sich der Alarm im Status `ALARM` befindet. Das Amazon-SNS-Thema kann Ihre E-Mail-Adresse enthalten,

sodass Sie eine E-Mail erhalten, wenn der Rechnungsbetrag den von Ihnen angegebenen Schwellenwert überschreitet.

Sie können ein vorhandenes Amazon-SNS-Thema auswählen, ein neues Amazon-SNS-Thema erstellen oder einen Themen-ARN verwenden, um ein anderes Konto zu benachrichtigen. Wenn Sie mehrere Benachrichtigungen für den gleichen Alarmstatus oder für verschiedene Alarm-Statuswerte senden möchten, wählen Sie Add notification (Benachrichtigung hinzufügen) aus.

14. Wählen Sie Weiter aus.
15. Geben Sie Name and description (Namen und Beschreibung) für Ihren Alarm ein. Der Name darf nur UTF-8-Zeichen und keine ASCII-Kontrolleingabezeichen enthalten.

    - (Optional) Geben Sie eine Beschreibung für Ihren Alarm ein. Die Beschreibung kann Markdown-Formatierungen enthalten, die nur auf der Registerkarte Alarmdetails in der CloudWatch Konsole angezeigt werden. Der Markdown kann nützlich sein, um Links zu Runbooks oder anderen internen Ressourcen hinzuzufügen.

16. Wählen Sie Weiter aus.
17. Vergewissern Sie sich unter Preview and create (Vorschau und Erstellung), ob Ihre Konfiguration korrekt ist, und wählen Sie Create alarm (Alarm erstellen) aus.

Löschen eines Abrechnung-Alarms

Sie können den Fakturierungsalarm löschen, wenn Sie ihn nicht mehr benötigen.

So löschen Sie einen Fakturierungsalarm

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Ändern Sie, falls erforderlich, die Region zu USA Ost (Nord-Virginia). Die metrischen Fakturierungsdaten werden in dieser Region gespeichert und stellen die weltweiten Gebühren dar.
3. Wählen Sie im Navigationsbereich Alarms (Alarme) und All alarms (Alle Alarme) aus.
4. Aktivieren Sie das Kontrollkästchen neben dem Alarm und wählen Sie Actions (Aktionen) und Delete (Löschen) aus.
5. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Ja, löschen.

Einen Alarm für die CPU-Auslastung erstellen

Sie können einen CloudWatch Alarm erstellen, der über Amazon SNS eine Benachrichtigung sendet, wenn der Status des Alarms von `OK` zu `ALARM` wechselt.

Der Alarm wechselt in den Status `ALARM`, wenn die durchschnittliche CPU-Auslastung einer EC2-Instance für aufeinanderfolgende angegebene Zeiträume einen bestimmten Schwellenwert überschreitet.

Einrichten eines Alarms für die CPU-Auslastung mithilfe des AWS Management Console

Gehen Sie wie folgt vor, um AWS Management Console mit dem einen Alarm zur CPU-Auslastung zu erstellen.

So erstellen Sie einen Alarm basierend auf der CPU-Auslastung

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich zuerst Alarme und dann Alle Alarme aus.
3. Wählen Sie Create alarm (Alarm erstellen) aus.
4. Wählen Sie Select metric (Metrik auswählen) aus.
5. Wählen Sie auf der Registerkarte Alle Metriken die Option EC2-Metriken.
6. Wählen Sie eine Metrikkategorie (z. B. Metriken pro Instance) aus.
7. Suchen Sie die Zeile mit der Instance, die Sie in der InstanceIdSpalte auflisten möchten, und nach CPUUtilization in der Spalte Metric Name. Aktivieren Sie das Kontrollkästchen neben dieser Zeile, und wählen Sie Metrik auswählen.
8. Wählen Sie unter Metrik und Bedingungen angeben für Statistik die Option Durchschnitt aus, wählen Sie eines der vordefinierten Perzentile aus oder geben Sie ein benutzerdefiniertes Perzentil an (z. B. **`p95.45`**).
9. Wählen Sie einen Zeitraum (z. B. **`5 minutes`**).
10. Geben Sie unter Conditions (Bedingungen) Folgendes an:

    a. Wählen Sie für Threshold type (Schwellenwerttyp) die Option Static (Statisch) aus.

    b. Geben Sie für Wann immer CPUUUtilization folgendes ist den Wert Größer an. Geben Sie unter als... den Schwellenwert an, der den Wechsel des Alarms in den ALARM-Zustand auslösen soll, wenn die CPU-Auslastung diesen Prozentsatz überschreitet. Beispiel: 70.

c. Wählen Sie Additional configuration (Zusätzliche Konfiguration). Geben Sie unter Datapoints to alarm (Datenpunkte für Alarm) an, wie viele Auswertungszeiträume (Datenpunkte) im Status `ALARM` sein müssen, damit der Alarm ausgelöst wird. Wenn die beiden Werte hier übereinstimmen, erstellen Sie einen Alarm, der in den Status `ALARM` wechselt, wenn entsprechend viele aufeinanderfolgende Zeiträume überschritten werden.

   Um einen M aus N Alarm zu erstellen, geben Sie eine niedrigere Zahl für den ersten Wert als für den zweiten Wert an. Weitere Informationen finden Sie unter Auswerten eines Alarms.

d. Wählen Sie für Missing data treatment (Behandlung von fehlenden Daten) aus, wie sich der Alarm verhalten soll, wenn einige Datenpunkte fehlen. Weitere Informationen finden Sie unter Konfiguration, wie Alarme fehlende Daten behandeln CloudWatch .

e. Wenn der Alarm ein Perzentil als überwachte Statistik verwendet, erscheint ein Feld Percentiles with low samples (Perzentile mit geringen Stichproben). Verwenden Sie diese Option, um zu entscheiden, ob Sie Fälle mit niedrigem Stichprobenumfang bewerten oder ignorieren möchten. Wenn Sie ignore (maintain alarm state) (Ignorieren (Alarmstatus beibehalten)) wählen, wird der aktuelle Alarmstatus immer beibehalten, wenn die Stichprobengröße zu gering ist. Weitere Informationen finden Sie unter Auf Perzentilen basierende CloudWatch Alarme und Stichproben mit niedrigen Datenmengen.

11. Wählen Sie Weiter.
12. Wählen Sie unter Benachrichtigung die Option In Alarm aus und wählen Sie ein SNS-Thema aus, das benachrichtigt werden soll, wenn sich der Alarm im Status `ALARM` befindet.

    Um zu erreichen, dass der Alarm mehrere Benachrichtigungen für den gleichen Alarmstatus oder für verschiedene Statuswerte sendet, wählen Sie Benachrichtigung hinzufügen.

    Damit der Alarm keine Benachrichtigungen sendet, wählen Sie Remove (Entfernen).
13. Wenn Sie fertig sind, wählen Sie Weiter.
14. Geben Sie einen Namen und eine Beschreibung für den Alarm ein. Wählen Sie anschließend Weiter.

    Der Name darf nur UTF-8-Zeichen und keine ASCII-Kontrolleingabezeichen enthalten. Die Beschreibung kann Markdown-Formatierungen enthalten, die nur auf der Registerkarte Alarmdetails in der Konsole angezeigt werden. CloudWatch Der Markdown kann nützlich sein, um Links zu Runbooks oder anderen internen Ressourcen hinzuzufügen.

15. Bestätigen Sie unter Preview and create (Vorschau und erstellen), dass die Informationen und Bedingungen den Anforderungen entsprechen, und wählen Sie dann Create alarm (Alarm erstellen).

Einrichten eines Alarms zur CPU-Auslastung mithilfe des AWS CLI

Gehen Sie wie folgt vor, um AWS CLI mit dem einen Alarm zur CPU-Auslastung zu erstellen.

So erstellen Sie einen Alarm basierend auf der CPU-Auslastung

1. Richten Sie ein SNS-Thema ein. Weitere Informationen finden Sie unter Einrichten von Amazon-SNS-Benachrichtigungen.
2. Erstellen Sie mit dem put-metric-alarmfolgenden Befehl einen Alarm.

```
aws cloudwatch put-metric-alarm --alarm-name cpu-mon --alarm-description "Alarm
 when CPU exceeds 70%" --metric-name CPUUtilization --namespace AWS/EC2 --statistic
 Average --period 300 --threshold 70 --comparison-operator GreaterThanThreshold --
dimensions  Name=InstanceId,Value=i-12345678 --evaluation-periods 2 --alarm-actions
 arn:aws:sns:us-east-1:111122223333:my-topic --unit Percent
```

3. Testen Sie den Alarm, indem Sie mit dem set-alarm-stateBefehl eine Änderung des Alarmstatus erzwingen.

   a. Ändern Sie den Alarmstatus von `INSUFFICIENT_DATA` in `OK`.

   ```
   aws cloudwatch set-alarm-state --alarm-name cpu-mon --state-reason
    "initializing" --state-value OK
   ```

   b. Ändern Sie den Alarmstatus von `OK` in `ALARM`.

   ```
   aws cloudwatch set-alarm-state --alarm-name cpu-mon --state-reason
    "initializing" --state-value ALARM
   ```

   c. Stellen Sie sicher, dass Sie eine Benachrichtigung über den Alarm erhalten haben.

Einen Load-Balancer-Latenz-Alarm erstellen, der E-Mails versendet

Sie können für den Classic Load Balancer eine Amazon-SNS-Benachrichtigung einrichten und einen Alarm konfigurieren, der überwacht, ob die Latenz 100 ms übersteigt.

Einrichten eines Latenzalarms mit dem AWS Management Console

Gehen Sie wie folgt vor, um einen Latenzalarm AWS Management Console für den Load Balancer zu erstellen.

Erstellen eines Load Balancer-Latenz-Alarms

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich zuerst Alarme und dann Alle Alarme aus.
3. Wählen Sie Create alarm (Alarm erstellen) aus.
4. Wählen Sie unter CloudWatch Metriken nach Kategorie die Kategorie ELB-Metriken aus.
5. Wählen Sie die Zeile mit Classic Load Balancer und der Metrik Latenz aus.
6. Für die Statistiken wählen Sie Average und dann eines der vordefinierten Perzentile aus, oder Sie geben ein benutzerdefiniertes Perzentil (z. B. **`p95.45`**) an.
7. Wählen Sie als Zeitraum 1 Minute aus.
8. Wählen Sie Weiter aus.
9. Geben Sie unter Alarm threshold (Alarmschwellwert) einen eindeutigen Namen für den Alarm (z. B. **`myHighCpuAlarm`**) und eine Beschreibung des Alarms (z. B. **`Alarm when Latency exceeds 100s`**) ein. Alarmnamen dürfen nur UTF-8-Zeichen und keine ASCII-Kontrolleingabezeichen enthalten

   Der Name darf nur UTF-8-Zeichen und keine ASCII-Kontrolleingabezeichen enthalten. Die Beschreibung kann eine Markdown-Formatierung enthalten, die nur auf der Registerkarte Alarmdetails in der CloudWatch Konsole angezeigt wird. Der Markdown kann nützlich sein, um Links zu Runbooks oder anderen internen Ressourcen hinzuzufügen.
10. Wählen Sie unter Whenever (Wenn) für is (ist) **>** und geben Sie **`0.1`** ein. Geben Sie für for (für) **`3`** ein.
11. Wählen Sie unter Additional settings (Zusätzliche Einstellungen) für Treat missing data as (Fehlende Daten behandeln als) ignore (maintain alarm state) (ignorieren (Alarmstatus beibehalten)), damit fehlende Datenpunkte keine Alarmstatusänderungen auslösen.

    Wählen Sie für Percentiles with low samples die Option ignore (maintain the alarm state) (ignorieren (Alarmzustand beibehalten)), sodass der Alarm nur Situationen mit ausreichender Anzahl von Datenbeispielen auswertet.

12. Wählen Sie unter Actions (Aktionen) für Whenever this alarm die Option State is ALARM (Status ist ALARM) aus. Wählen Sie für Send notification to ein vorhandenes SNS-Thema aus oder erstellen Sie ein neues.

    Um ein neues SNS-Thema zu erstellen, wählen Sie New list aus. Geben Sie für Send notification to (Benachrichtigung senden an) einen Namen für das SNS-Thema (z. B. **myHighCpuAlarm**) und für Email list (E-Mail-Liste) eine kommagetrennte Liste von E-Mail-Adressen ein, die benachrichtigt werden sollen, wenn der Alarm in den Status `ALARM` wechselt. Jeder E-Mail-Adresse wird ein Bestätigungs-E-Mail für das Abonnement eines Themas gesendet. Sie müssen das Abonnement bestätigen, bevor Benachrichtigungen gesendet werden können.

13. Wählen Sie Alarm erstellen aus.

Einrichten eines Latenzalarms mit dem AWS CLI

Gehen Sie wie folgt vor, um einen Latenzalarm AWS CLI für den Load Balancer zu erstellen.

Erstellen eines Load-Balancer-Latenz-Alarms

1. Richten Sie ein SNS-Thema ein. Weitere Informationen finden Sie unter Einrichten von Amazon-SNS-Benachrichtigungen.
2. Erstellen Sie den Alarm mit dem folgenden put-metric-alarmBefehl:

```
aws cloudwatch put-metric-alarm --alarm-name lb-mon --alarm-description "Alarm
 when Latency exceeds 100s" --metric-name Latency --namespace AWS/ELB --statistic
 Average --period 60 --threshold 100 --comparison-operator GreaterThanThreshold --
dimensions Name=LoadBalancerName,Value=my-server --evaluation-periods 3 --alarm-
actions arn:aws:sns:us-east-1:111122223333:my-topic --unit Seconds
```

3. Testen Sie den Alarm, indem Sie mit dem set-alarm-stateBefehl eine Änderung des Alarmstatus erzwingen.

    a. Ändern Sie den Alarmstatus von `INSUFFICIENT_DATA` in `OK`.

    ```
    aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason
     "initializing" --state-value OK
    ```

    b. Ändern Sie den Alarmstatus von `OK` in `ALARM`.

```
aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason
 "initializing" --state-value ALARM
```

c. Stellen Sie sicher, dass Sie eine E-Mail-Benachrichtigung über den Alarm erhalten haben.

Einen Speicherdurchsatzalarm erstellen, der E-Mails versendet

Sie können eine SNS-Benachrichtigung einrichten und einen Alarm konfigurieren, der E-Mails versendet, wenn der Amazon-EBS-Durchsatz 100 MB überschreitet.

Einrichten eines Speicherdurchsatz-Alarms mit AWS Management Console

Gehen Sie wie folgt vor, um einen Alarm AWS Management Console zu erstellen, der auf dem Amazon EBS-Durchsatz basiert.

Erstellen eines Speicherdurchsatzalarms

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich zuerst Alarme und dann Alle Alarme aus.
3. Wählen Sie Create alarm (Alarm erstellen) aus.
4. Wählen Sie unter EBS Metrics eine Metrikkategorie aus.
5. Wählen Sie die Zeile mit dem Volumen und der VolumeWriteBytesMetrik aus.
6. Wählen Sie im Bereich Statistik die Option Average aus. Wählen Sie als Zeitraum 5 Minuten aus. Wählen Sie Weiter aus.
7. Geben Sie unter Alarm threshold (Alarmschwellwert) einen eindeutigen Namen für den Alarm (z. B. **myHighWriteAlarm**) und eine Beschreibung des Alarms (z. B. **VolumeWriteBytes exceeds 100,000 KiB/s**) ein. Der Name darf nur UTF-8-Zeichen und keine ASCII-Kontrolleingabezeichen enthalten. Die Beschreibung kann eine Markdown-Formatierung enthalten, die nur auf der Registerkarte „Alarmdetails“ in der CloudWatch Konsole angezeigt wird. Der Markdown kann nützlich sein, um Links zu Runbooks oder anderen internen Ressourcen hinzuzufügen.
8. Wählen Sie unter Whenever (Wenn) für is (ist) **>** und geben Sie **100000** ein. Geben Sie für for (für) **15** aufeinanderfolgende Zeiträume ein.

   Eine grafische Darstellung dieses Schwellenwerts ist unter Alarm Preview zu finden.

9. Wählen Sie unter Additional settings (Zusätzliche Einstellungen) für Treat missing data as (Fehlende Daten behandeln als) ignore (maintain alarm state) (ignorieren (Alarmstatus beibehalten)), damit fehlende Datenpunkte keine Alarmstatusänderungen auslösen.
10. Wählen Sie unter Actions (Aktionen) für Whenever this alarm die Option State is ALARM (Status ist ALARM) aus. Wählen Sie für Send notification to (Benachrichtigung senden an) ein vorhandenes SNS-Thema aus oder erstellen Sie ein neues.

    Um ein neues SNS-Thema zu erstellen, wählen Sie New list aus. Geben Sie für Send notification to (Benachrichtigung senden an) einen Namen für das SNS-Thema (z. B. **`myHighCpuAlarm`**) und für Email list (E-Mail-Liste) eine kommagetrennte Liste von E-Mail-Adressen ein, die benachrichtigt werden sollen, wenn der Alarm in den Status `ALARM` wechselt. Jeder E-Mail-Adresse wird ein Bestätigungs-E-Mail für das Abonnement eines Themas gesendet. Sie müssen das Abonnement bestätigen, bevor Benachrichtigungen an eine E-Mail-Adresse gesendet werden können.
11. Wählen Sie Alarm erstellen aus.

Einrichten eines Alarms für den Speicherdurchsatz mithilfe des AWS CLI

Gehen Sie wie folgt vor, um einen Alarm AWS CLI zu erstellen, der auf dem Amazon EBS-Durchsatz basiert.

Erstellen eines Speicherdurchsatzalarms

1. Erstellen Sie ein SNS-Thema. Weitere Informationen finden Sie unter Einrichten von Amazon-SNS-Benachrichtigungen.
2. Erstellen Sie den Alarm.

```
aws cloudwatch put-metric-alarm --alarm-name ebs-mon --alarm-description "Alarm
 when EBS volume exceeds 100MB throughput" --metric-name VolumeReadBytes --
namespace AWS/EBS --statistic Average --period 300 --threshold 100000000 --
comparison-operator GreaterThanThreshold --dimensions Name=VolumeId,Value=my-
volume-id --evaluation-periods 3 --alarm-actions arn:aws:sns:us-
east-1:111122223333:my-alarm-topic --insufficient-data-actions arn:aws:sns:us-
east-1:111122223333:my-insufficient-data-topic
```

3. Testen Sie den Alarm, indem Sie mit dem set-alarm-stateBefehl eine Änderung des Alarmstatus erzwingen.

    a. Ändern Sie den Alarmstatus von `INSUFFICIENT_DATA` in `OK`.

```
aws cloudwatch set-alarm-state --alarm-name ebs-mon --state-reason
 "initializing" --state-value OK
```

b. Ändern Sie den Alarmstatus von `OK` in `ALARM`.

```
aws cloudwatch set-alarm-state --alarm-name ebs-mon --state-reason
 "initializing" --state-value ALARM
```

c. Ändern Sie den Alarmstatus von `ALARM` in `INSUFFICIENT_DATA`.

```
aws cloudwatch set-alarm-state --alarm-name ebs-mon --state-reason
 "initializing" --state-value INSUFFICIENT_DATA
```

d. Stellen Sie sicher, dass Sie eine E-Mail-Benachrichtigung über den Alarm erhalten haben.

Einen Alarm für Performance Insights Insights-Zählermetriken aus einer AWS Datenbank erstellen

CloudWatch enthält eine metrische Datenbankfunktion DB_PERF_INSIGHTS, mit der Sie Performance Insights Insights-Zählermetriken CloudWatch aus Amazon Relational Database Service und Amazon DocumentDB (mit MongoDB-Kompatibilität) importieren können. DB_PERF_INSIGHTS fügt die `DBLoad`-Metrik auch in Intervallen unter einer Minute ein. Sie können Alarme für diese Metriken einrichten. CloudWatch

Weitere Informationen über Erkenntnisse zur Amazon-RDS-Leistung finden Sie unter Überwachung der DB-Auslastung mit Performance Insights auf Amazon RDS.

Weitere Informationen über Amazon DocumentDB Performance Insights finden Sie unter Überwachen mit Performance Insights.

Die Anomalieerkennung wird für Alarme, die auf der Funktion DB_PERF_INSIGHTS basieren, nicht unterstützt.

> **Note**
>
> Hochauflösende Metriken mit Subminuten-Granularität, die von DB_PERF_INSIGHTS abgerufen werden, gelten nur für die DBLoad-Metrik oder für Betriebssystem-Metriken, wenn Sie Enhanced Monitoring mit einer höheren Auflösung aktiviert haben. Weitere

Informationen zur erweiterten Überwachung von Amazon RDS finden Sie unter Überwachen von Betriebssystemmetriken mit Enhanced Monitoring.
Mit der Funktion DB_PERF_INSIGHTS können Sie einen Alarm mit hoher Auflösung erstellen. Der maximale Bewertungsbereich für einen Alarm mit hoher Auflösung beträgt drei Stunden. Sie können die CloudWatch Konsole verwenden, um mit der Funktion DB_PERF_INSIGHTS abgerufene Metriken für einen beliebigen Zeitraum grafisch darzustellen.

So erstellen Sie einen Alarm, der auf Performance-Insights-Metriken basiert

1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/cloudwatch/ CloudWatch .
2. Wählen Sie im Navigationsbereich Alarme und dann Alle Alarme aus.
3. Wählen Sie Create alarm (Alarm erstellen) aus.
4. Wählen Sie Select Metric (Metrik auswählen) aus.
5. Wählen Sie das Dropdown-Menü Add math und wählen Sie dann Database Performance Metrics, DB_PERF_INSIGHTS aus der Liste aus.

   Nachdem Sie DB_PERF_INSIGHTS ausgewählt haben, erscheint ein Feld für mathematische Ausdrücke, in dem Sie mathematische Ausdrücke anwenden oder bearbeiten können.
6. Geben Sie in das Feld mathematischer Ausdruck Ihren mathematischen Ausdruck DB_PERF_INSIGHTS ein und wählen Sie dann Anwenden.

   Beispiel: **`DB_PERF_INSIGHTS('RDS', 'db-ABCDEFGHIJKLMNOPQRSTUVWXY1', 'os.cpuUtilization.user.avg')`**

   > **Important**
   >
   > Wenn Sie den mathematischen Ausdruck DB_PERF_INSIGHTS verwenden, müssen Sie die eindeutige Datenbankressourcen-ID der Datenbank angeben. Dies unterscheidet sich von der Datenbank-ID. Um die Datenbank-Ressourcen-ID in der Amazon-RDS-Konsole zu finden, wählen Sie die DB-Instance, um ihre Details anzuzeigen. Wechseln Sie zur Registerkarte Konfiguration. Die Ressourcen-ID wird im Abschnitt Konfiguration angezeigt.

Informationen über die Funktion DB_PERF_INSIGHTS und andere Funktionen, die für Metrikberechnungen verfügbar sind, finden Sie unter Syntax und Funktionen von Metrikberechnungen.

7. Wählen Sie Select metric (Metrik auswählen) aus.

   Die Seite Specify metric and conditions (Metrik und Bedingungen festlegen) wird angezeigt, die ein Diagramm und andere Informationen über den von Ihnen ausgewählten mathematischen Ausdruck anzeigt.

8. Geben Sie für Wann immer der ***Ausdruck*** ist an, ob der Ausdruck größer, kleiner oder gleich dem Schwellenwert sein muss. Geben Sie unter than... (dann ...) den Schwellenwert an.

9. Wählen Sie Additional configuration (Zusätzliche Konfiguration). Geben Sie unter Datapoints to alarm (Datenpunkte für Alarm) an, wie viele Auswertungszeiträume (Datenpunkte) im Status `ALARM` sein müssen, damit der Alarm ausgelöst wird. Wenn die beiden Werte hier übereinstimmen, erstellen Sie einen Alarm, der in den Status `ALARM` wechselt, wenn entsprechend viele aufeinanderfolgende Zeiträume überschritten werden.

   Um einen M aus N Alarm zu erstellen, geben Sie eine niedrigere Zahl für den ersten Wert als für den zweiten Wert an. Weitere Informationen finden Sie unter Auswerten eines Alarms.

10. Wählen Sie für Missing data treatment (Behandlung von fehlenden Daten) aus, wie sich der Alarm verhalten soll, wenn einige Datenpunkte fehlen. Weitere Informationen finden Sie unter Konfiguration, wie Alarme fehlende Daten behandeln CloudWatch .

11. Wählen Sie Weiter.

12. Wählen Sie unter Notification (Benachrichtigung) ein SNS-Thema aus, das benachrichtigt werden soll, wenn sich der Alarm im Status `ALARM`, `OK` oder `INSUFFICIENT_DATA` befindet.

    Um zu erreichen, dass der Alarm mehrere Benachrichtigungen für den gleichen Alarmstatus oder für verschiedene Statuswerte sendet, wählen Sie Benachrichtigung hinzufügen.

    Damit der Alarm keine Benachrichtigungen sendet, wählen Sie Remove (Entfernen).

13. Um den Alarm Auto-Scaling-, EC2-, Lambda- oder Systems-Manager-Aktionen ausführen zu lassen, wählen Sie die entsprechende Schaltfläche und wählen Sie den Alarmstatus und die auszuführende Aktion. Wenn Sie eine Lambda-Funktion als Alarmaktion wählen, geben Sie den Funktionsnamen oder ARN an, und Sie können optional eine bestimmte Version der Funktion auswählen.

Alarme können Aktionen des Systems Manager nur ausführen, wenn sie in den ALARM-Zustand wechseln. Weitere Informationen zu Systems Manager Manager-Aktionen finden Sie unter Konfiguration für CloudWatch die Erstellung OpsItems aus Alarmen und Incident-Erstellung.

> **Note**
>
> Um einen Alarm zu erstellen, der eine SSM-Incident-Manager-Aktion ausführt, müssen Sie über bestimmte Berechtigungen verfügen. Weitere Informationen finden Sie unter Beispiele für identitätsbasierte Richtlinien für AWS Systems Manager Incident Manager.

14. Wenn Sie fertig sind, wählen Sie Weiter.
15. Geben Sie einen Namen und eine Beschreibung für den Alarm ein. Wählen Sie anschließend Weiter.

    Der Name darf nur UTF-8-Zeichen und keine ASCII-Kontrolleingabezeichen enthalten. Die Beschreibung kann Markdown-Formatierungen enthalten, die nur auf der Registerkarte Alarmdetails in der Konsole angezeigt werden. CloudWatch Der Markdown kann nützlich sein, um Links zu Runbooks oder anderen internen Ressourcen hinzuzufügen.
16. Bestätigen Sie unter Preview and create (Vorschau und erstellen), dass die Informationen und Bedingungen den Anforderungen entsprechen, und wählen Sie dann Create alarm (Alarm erstellen).

Erstellen Sie Alarme, um eine EC2-Instance anzuhalten, zu beenden, neu zu starten oder wiederherzustellen

Mithilfe von CloudWatch Amazon-Alarmaktionen können Sie Alarme erstellen, die Ihre EC2-Instances automatisch stoppen, beenden, neu starten oder wiederherstellen. Sie können die Aktionen zum Anhalten oder Beenden nutzen, um Geld zu sparen, wenn eine Instance über einen längeren Zeitraum nicht ausgeführt werden muss. Sie können die Aktionen zum Neustarten oder Wiederherstellen verwenden, um diese Instances automatisch neu zu starten oder um sie – für den Fall, dass eine Systembeeinträchtigung eintritt – auf einer neuen Hardware wiederherzustellen.

Es gibt eine Reihe von Szenarien, bei denen Sie Ihre Instance möglicherweise automatisch anhalten oder beenden möchten. Beispielsweise verwenden Sie Instances für die Batchverarbeitung von Gehaltsabrechnungen oder wissenschaftliche Datenverarbeitungsaufgaben, die für einen bestimmten Zeitraum ausgeführt werden und ihre Arbeit anschließend abschließen. Anstatt diese Instances im

Leerlauf beizubehalten (und damit Kosten anfallen zu lassen), können Sie sie auch anhalten oder beenden und so Geld sparen. Der Hauptunterschied zwischen der Verwendung einer Alarmaktion zum Anhalten und einer Alarmaktion zum Beenden besteht darin, dass Sie eine angehaltene Instance problemlos wieder neu starten können, wenn sie später wieder ausgeführt werden soll. Sie können auch die gleiche Instance-ID und das gleiche Stamm-Volume beibehalten. Eine beendete Instance können Sie dagegen nicht neu starten. Stattdessen müssen Sie eine neue Instance starten.

Sie können die Aktionen Stoppen, Beenden oder Neustarten zu jedem Alarm hinzufügen, der für eine Amazon EC2-Metrik pro Instance festgelegt ist, einschließlich grundlegender und detaillierter Überwachungsmetriken, die von Amazon CloudWatch (im AWS/EC2-Namespace) bereitgestellt werden, zusätzlich zu allen benutzerdefinierten Metriken, die die Dimension "InstanceId=“ enthalten, sofern sich der InstanceId Wert auf eine gültige laufende Amazon EC2 EC2-Instance bezieht. Sie können die Wiederherstellungsaktion auch zu Alarmen hinzufügen, die für jede Amazon EC2-Metrik pro Instance festgelegt sind, mit Ausnahme von `StatusCheckFailed_Instance`.

Um eine CloudWatch Alarmaktion einzurichten, mit der eine Instance neu gestartet, gestoppt oder beendet werden kann, müssen Sie eine serviceverknüpfte IAM-Rolle verwenden, AWSServiceRoleForCloudWatchEvents. Die AWSServiceRoleForCloudWatchEvents IAM-Rolle ermöglicht es AWS , Alarmaktionen in Ihrem Namen durchzuführen.

Verwenden Sie den folgenden Befehl, um die serviceverknüpfte Rolle für CloudWatch Ereignisse zu erstellen:

```
aws iam create-service-linked-role --aws-service-name events.amazonaws.com
```

Konsolenunterstützung

Sie können Alarme mit der CloudWatch Konsole oder der Amazon EC2 EC2-Konsole erstellen. Die Verfahren in dieser Dokumentation verwenden die CloudWatch Konsole. Weitere Informationen zu Prozeduren, die die Amazon-EC2-Konsole verwenden, finden Sie unter Erstellen von Alarmen, mit denen eine Instance angehalten, beendet, neu gestartet oder wiederhergestellt wird in Amazon-EC2-Benutzerhandbuch für Linux-Instances.

Berechtigungen

Wenn Sie ein AWS Identity and Access Management (IAM-) Konto verwenden, um einen Alarm zu erstellen oder zu ändern, der EC2-Aktionen oder Systems Manager OpsItem Manager-Aktionen ausführt, benötigen Sie die `iam:CreateServiceLinkedRole` entsprechende Genehmigung.

Inhalt

Hinzufügen von Stoppaktionen zu CloudWatch Amazon-Alarmen

Sie können einen Alarm erstellen, mit dem eine Amazon EC2- Instance angehalten wird, sobald ein bestimmter Schwellenwert erreicht wird. Es kann beispielsweise sein, dass Sie Entwicklungs- oder Test-Instances ausführen und gelegentlich vergessen, diese herunterzufahren. Sie können einen Alarm einrichten, der ausgelöst wird, wenn die durchschnittliche prozentuale CPU-Auslastung 24 Stunden lang unter 10 Prozent fällt. Dies signalisiert, dass sich die Instance im Leerlauf befindet und nicht mehr verwendet wird. Sie können den Schwellenwert, die Dauer und den Zeitraum an Ihre Anforderungen anpassen. Außerdem haben Sie die Möglichkeit, eine SNS-Benachrichtigung hinzuzufügen, damit Sie eine E-Mail erhalten,sobald der Alarm ausgelöst wird.

Amazon-EC2-Instances, die ein Volume von Amazon Elastic Block Store als Root-Gerät verwenden, können angehalten oder beendet werden. Instances, die den Instance-Speicher als Root-Gerät verwenden, können dagegen nur beendet werden.

Um mithilfe der CloudWatch Amazon-Konsole einen Alarm zum Stoppen einer Instance im Leerlauf zu erstellen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Alarms (Alarme) und All alarms (Alle Alarme) aus.
3. Wählen Sie Create alarm (Alarm erstellen) aus.
4. Wählen Sie Select Metric (Metrik auswählen) aus.
5. Wählen Sie für AWS -Namespaces die Option EC2 aus.
6. Gehen Sie wie folgt vor:

    a. Wählen Sie Per-Instance Metrics (Metriken pro Instance) aus.
    b. Wählen Sie das Kontrollkästchen in der Zeile mit der richtigen Instance und der Metrik CPUUtilization aus.
    c. Wählen Sie die Registerkarte Graphed metrics (Grafisch dargestellte Metriken) aus.
    d. Wählen Sie im Bereich Statistik die Option Average aus.

e. Wählen Sie einen Zeitraum (z. B. **1 Hour**).

f. Wählen Sie Select metric (Metrik auswählen) aus.

7. Führen Sie für den Schritt Define Alarm Folgendes aus:

a. Wählen Sie unter Conditions (Bedingungen) die Option Static (Statisch) aus.

b. Wählen Sie unter Whenever CPUUtilization is (Wenn CPUUtilization ist) die Option Lower (Kleiner) aus.

c. Geben Sie für than (als) **10** ein.

d. Wählen Sie Weiter aus.

e. Wählen Sie unter Notification für Send notification to ein vorhandenes SNS-Thema aus oder erstellen Sie ein neues.

Um ein neues SNS-Thema zu erstellen, wählen Sie New list aus. Geben Sie unter Send a notification to (Benachrichtigung senden an) einen Namen für das SNS-Thema ein (z. B. EC2-Instance beenden). Geben Sie für Email list (E-Mail-Liste) eine durch Kommata getrennte Liste der E-Mail-Adressen ein, die benachrichtigt werden sollen, wenn der Alarm in den Status ALARM versetzt wird. Jeder E-Mail-Adresse wird ein Bestätigungs-E-Mail für das Abonnement eines Themas gesendet. Sie müssen das Abonnement bestätigen, bevor Benachrichtigungen an eine E-Mail-Adresse gesendet werden können.

f. Wählen Sie Add EC2 Action (EC2-Aktion hinzufügen) aus.

g. Wählen Sie für Alarmstatusauslöser die Option Im Alarm aus. Wählen Sie für Folgende Aktion ausführen die Option Diese Instance stoppen aus.

h. Wählen Sie Weiter aus.

i. Geben Sie einen Namen und eine Beschreibung für den Alarm ein. Der Name darf nur ASCII-Zeichen enthalten. Wählen Sie anschließend Weiter.

j. Bestätigen Sie unter Preview and create (Vorschau und erstellen), dass die Informationen und Bedingungen den Anforderungen entsprechen, und wählen Sie dann Create alarm (Alarm erstellen).

Abbruchaktionen zu CloudWatch Amazon-Alarmen hinzufügen

Sie können einen Alarm erstellen, mit dem eine EC2 Instance automatisch beendet wird, sobald ein bestimmter Schwellenwert erreicht wird (solange für die Instance kein Beendigungsschutz aktiviert ist). Möglicherweise möchten Sie z. B. eine Instance beenden, sobald sie ihre Arbeit abgeschlossen hat, und Sie benötigen die Instance nicht erneut. Wenn Sie die Instance unter

Umständen später noch einmal verwenden möchten, sollten Sie die Instance nur anhalten, anstatt sie zu beenden. Weitere Informationen zur Aktivierung und Deaktivierung des Beendigungsschutzes für eine Instance finden Sie unter Aktivieren des Beendigungsschutzes für eine Instance im Amazon-EC2-Benutzerhandbuch für Linux-Instances.

Um einen Alarm zu erstellen, um eine Instance im Leerlauf mithilfe der CloudWatch Amazon-Konsole zu beenden

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Alarms und Alarm erstellen aus.
3. Führen Sie für den Schritt Select Metric (Metrik auswählen) Folgendes aus:
   a. Wählen Sie unter EC2 Metrics die Option Per-Instance Metrics aus.
   b. Wählen Sie die Zeile mit der Instance und der Metrik CPUUtilization aus.
   c. Wählen Sie im Bereich Statistik die Option Average aus.
   d. Wählen Sie einen Zeitraum (z. B. **1 Hour**).
   e. Wählen Sie Weiter aus.
4. Führen Sie für den Schritt Define Alarm Folgendes aus:
   a. Geben Sie unter Alarm Threshold einen eindeutigen Namen für den Alarm (z. B. EC2-Instance beenden) und eine Beschreibung des Alarms ein (z. B. EC2-Instance beenden, sobald die CPU zu lange im Leerlauf ist). Alarmnamen dürfen nur ASCII-Zeichen enthalten.
   b. Wählen Sie unter Whenever (Wenn) für is (ist) die Option **<** und geben Sie **10** ein. Geben Sie für für **24** aufeinanderfolgende Zeiträume ein.

      Eine grafische Darstellung dieses Schwellenwerts ist unter Alarm Preview zu finden.
   c. Wählen Sie unter Notification für Send notification to ein vorhandenes SNS-Thema aus oder erstellen Sie ein neues.

      Um ein neues SNS-Thema zu erstellen, wählen Sie New list aus. Geben Sie unter Send a notification to (Benachrichtigung senden an) einen Namen für das SNS-Thema ein (z. B. EC2-Instance beenden). Geben Sie für Email list (E-Mail-Liste) eine durch Kommata getrennte Liste der E-Mail-Adressen ein, die benachrichtigt werden sollen, wenn der Alarm in den Status `ALARM` versetzt wird. Jeder E-Mail-Adresse wird ein Bestätigungs-E-Mail für das Abonnement eines Themas gesendet. Sie müssen das Abonnement bestätigen, bevor Benachrichtigungen an eine E-Mail-Adresse gesendet werden können.

d. Wählen Sie EC2 Action aus.

e. Wählen Sie für Whenever this alarm die Option State is ALARM aus. Wählen Sie für Take this action die Option Terminate this instance aus.

f. Wählen Sie Alarm erstellen aus.

Neustartaktionen zu CloudWatch Amazon-Alarmen hinzufügen

Sie können einen CloudWatch Amazon-Alarm erstellen, der eine Amazon EC2-Instance überwacht und die Instance automatisch neu startet. Die Alarmaktion zum Neustarten wird für Instance-Zustandsprüfungsfehler empfohlen (im Gegensatz zur Alarmaktion zum Wiederherstellen, die sich für System-Zustandsprüfungsfehler eignet). Ein Neustart einer Instance entspricht einem Neustart des Betriebssystems. In den meisten Fällen dauert es nur wenige Minuten, um die Instance neu zu starten. Wenn Sie eine Instance neu starten, verbleibt sie auf demselben physischen Host, sodass die Instance ihren öffentlichen DNS- Namen, ihre private IP-Adresse sowie alle Daten auf ihren Instance-Speicher-Volumes behält.

Im Gegensatz zum Anhalten oder Neustarten der Instance beginnt mit dem erneuten Hochfahren einer Instance nicht eine neue Instance-Abrechnungsstunde. Weitere Informationen zum Neustarten einer Instance finden Sie unter Instance erneut starten im Amazon-EC2-Benutzerhandbuch für Linux-Instances.

> **Important**
>
> Um eine Racebedingung zwischen der Neustart- und der Wiederherstellungsaktion zu vermeiden, sollten Sie für den Neustartalarm und den Wiederherstellungsalarm nicht den gleichen Auswertungszeitraum festlegen. Wir empfehlen, dass Sie Neustartalarme zu drei Auswertungszeiträumen von jeweils einer Minute festlegen.

Um einen Alarm für den Neustart einer Instance mithilfe der CloudWatch Amazon-Konsole zu erstellen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Alarms und Alarm erstellen aus.
3. Führen Sie für den Schritt Select Metric (Metrik auswählen) Folgendes aus:

   a. Wählen Sie unter EC2 Metrics die Option Per-Instance Metrics aus.

b. Wählen Sie die Zeile mit der Instanz und der Metrik StatusCheckFailed_Instance aus.

c. Wählen Sie für Statistik die Option Minimum aus.

d. Wählen Sie einen Zeitraum (z. B. **1 Minute**).

e. Wählen Sie Weiter aus.

4. Führen Sie für den Schritt Define Alarm Folgendes aus:

a. Geben Sie unter Alarm Threshold einen eindeutigen Namen für den Alarm (z. B. EC2-Instance neu starten) und eine Beschreibung des Alarms ein (z. B. EC2-Instance neu starten, sobald die Zustandsprüfung fehlschlägt). Alarmnamen dürfen nur ASCII-Zeichen enthalten.

b. Wählen Sie unter Whenever (Jederzeit) für ist **>** und geben Sie **0** ein. Geben Sie für für **3** aufeinanderfolgende Zeiträume ein.

Eine grafische Darstellung dieses Schwellenwerts ist unter Alarm Preview zu finden.

c. Wählen Sie unter Notification für Send notification to ein vorhandenes SNS-Thema aus oder erstellen Sie ein neues.

Um ein neues SNS-Thema zu erstellen, wählen Sie New list aus. Geben Sie unter Send a notification to (Benachrichtigung senden an) einen Namen für das SNS-Thema ein (z. B. EC2-Instance neu starten). Geben Sie für Email list (E-Mail-Liste) eine durch Kommata getrennte Liste der E-Mail-Adressen ein, die benachrichtigt werden sollen, wenn der Alarm in den Status `ALARM` versetzt wird. Jeder E-Mail-Adresse wird ein Bestätigungs-E-Mail für das Abonnement eines Themas gesendet. Sie müssen das Abonnement bestätigen, bevor Benachrichtigungen an eine E-Mail-Adresse gesendet werden können.

d. Wählen Sie EC2 Action aus.

e. Wählen Sie für Whenever this alarm die Option State is ALARM aus. Wählen Sie für Take this action die Option Reboot this instance aus.

f. Wählen Sie Alarm erstellen aus.

Wiederherstellungsaktionen zu CloudWatch Amazon-Alarmen hinzufügen

Sie können einen CloudWatch Amazon-Alarm erstellen, der eine Amazon EC2-Instance überwacht und die Instance automatisch wiederherstellt, wenn sie aufgrund eines zugrunde liegenden Hardwarefehlers oder eines Problems, das eine Reparatur erfordert AWS , beeinträchtigt wird. Beendete Instances können nicht wiederhergestellt werden. Eine wiederhergestellte Instance ist

mit der ursprünglichen Instance identisch. Dies schließt auch die Instance-ID, private IP-Adressen, Elastic IP-Adressen und alle Instance-Metadaten mit ein.

Wird der Alarm `StatusCheckFailed_System` ausgelöst und die Aktion zum Wiederherstellen initiiert, werden Sie über das Amazon-SNS-Thema, das Sie bei der Erstellung des Alarms gewählt haben und das mit der Aktion zum Wiederherstellen verknüpft ist, darüber benachrichtigt. Während der Instance-Wiederherstellung wird die Instance bei einem Instance-Neustart migriert und alle im Speicher befindlichen Daten gehen verloren. Wenn der Vorgang abgeschlossen ist, wird die Information in dem SNS-Thema, das Sie für den Alarm konfiguriert haben, veröffentlicht. Alle Personen, die das SNS-Thema abonniert haben, erhalten eine Benachrichtigung per E-Mail, in der auch der Status des Wiederherstellungsversuchs und weitere Anweisungen enthalten sind. Sie werden bemerken, dass auf der wiederhergestellten Instance ein Instance-Neustart durchgeführt wird.

Die Aktion zum Wiederherstellen kann nur mit `StatusCheckFailed_System` verwendet werden, nicht mit `StatusCheckFailed_Instance`.

Beispiele für Probleme, die dazu führen, dass Systemstatusprüfungen fehlschlagen, sind:

- Verlust der Netzwerkverbindung
- Systemstromausfall
- Softwareprobleme auf dem physischen Host
- Hardwareprobleme auf dem physischen Host, die die Erreichbarkeit des Netzwerks beeinträchtigen

Die Wiederherstellungsaktion wird nur auf einigen Instances unterstützt. Weitere Informationen zu unterstützten Instance-Typen und anderen Anforderungen finden Sie unter Ihre Instance wiederherstellen und Anforderungen.

> **Important**
>
> Um eine Racebedingung zwischen der Neustart- und der Wiederherstellungsaktion zu vermeiden, sollten Sie für den Neustartalarm und den Wiederherstellungsalarm nicht den gleichen Auswertungszeitraum festlegen. Wir empfehlen, dass Sie Wiederherstellungsalarme zu zwei Auswertungszeiträumen von jeweils einer Minute und Neustartalarme zu drei Auswertungszeiträumen von jeweils einer Minute festlegen.

Um einen Alarm zur Wiederherstellung einer Instance mithilfe der CloudWatch Amazon-Konsole zu erstellen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Alarms und Alarm erstellen aus.
3. Führen Sie für den Schritt Select Metric (Metrik auswählen) Folgendes aus:

   a. Wählen Sie unter EC2 Metrics die Option Per-Instance Metrics aus.

   b. Wählen Sie die Zeile mit der Instanz und der Metrik StatusCheckFailed_System aus.

   c. Wählen Sie für Statistik die Option Minimum aus.

   d. Wählen Sie einen Zeitraum (z. B. **`1 Minute`**).

   > **Important**
   >
   > Um eine Racebedingung zwischen der Neustart- und der Wiederherstellungsaktion zu vermeiden, sollten Sie für den Neustartalarm und den Wiederherstellungsalarm nicht den gleichen Auswertungszeitraum festlegen. Wir empfehlen, dass Sie Wiederherstellungsalarme zu zwei Auswertungszeiträumen von jeweils einer Minute festlegen.

   e. Wählen Sie Weiter aus.
4. Führen Sie für den Schritt Define Alarm Folgendes aus:

   a. Geben Sie unter Alarm Threshold einen eindeutigen Namen für den Alarm (z. B. EC2-Instance wiederherstellen) und eine Beschreibung des Alarms ein (z. B. EC2-Instance wiederherstellen, sobald die Zustandsprüfung fehlschlägt). Alarmnamen dürfen nur ASCII-Zeichen enthalten.

   b. Wählen Sie unter Whenever (Jederzeit) für ist **>** und geben Sie **`0`** ein. Geben Sie für für **`2`** aufeinanderfolgende Zeiträume ein.

   c. Wählen Sie unter Notification für Send notification to ein vorhandenes SNS-Thema aus oder erstellen Sie ein neues.

      Um ein neues SNS-Thema zu erstellen, wählen Sie New list aus. Geben Sie unter Send a notification to (Benachrichtigung senden an) einen Namen für das SNS-Thema ein (z. B. EC2-Instance wiederherstellen). Geben Sie für Email list (E-Mail-Liste) eine durch Kommata getrennte Liste der E-Mail-Adressen ein, die benachrichtigt werden sollen, wenn der Alarm in den Status `ALARM` versetzt wird. Jeder E-Mail-Adresse wird ein Bestätigungs-E-Mail für

das Abonnement eines Themas gesendet. Sie müssen das Abonnement bestätigen, bevor Benachrichtigungen an eine E-Mail-Adresse gesendet werden können.

d. Wählen Sie EC2 Action aus.

e. Wählen Sie für Whenever this alarm die Option State is ALARM aus. Wählen Sie für Take this action die Option Recover this instance aus.

f. Wählen Sie Alarm erstellen aus.

Verlauf ausgelöster Alarme und Aktionen ansehen

Sie können den Alarm- und Aktionsverlauf in der CloudWatch Amazon-Konsole einsehen. Amazon CloudWatch speichert den Alarm- und Aktionsverlauf der letzten 30 Tage.

So zeigen Sie den Verlauf ausgelöster Alarme und Aktionen an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Alarms (Alarme) und wählen Sie einen Alarm aus.
3. Wählen Sie die Registerkarte Details aus, um neben den Zeit- und Metrikwerten den neuesten Statusübergang anzuzeigen.
4. Wählen Sie die Registerkarte History (Verlauf) aus, um die neuesten Verlaufseinträge anzuzeigen.

Alarme und Tagging

Stichwörter sind Schlüssel-Wert-Paare, die Ihnen helfen können, Ihre Ressourcen zu organisieren und zu kategorisieren. Sie können sie außerdem verwenden, um Benutzer-Berechtigungen abzugrenzen. Dazu erteilen Sie einem Benutzer die Berechtigung für den Zugriff oder die Änderung von Ressourcen mit bestimmten Tag-Werten. Weitere allgemeine Informationen zum Markieren von Ressourcen finden Sie unter Ressourcen taggen AWS

In der folgenden Liste werden einige Details dazu erklärt, wie das Markieren von Alarmen funktioniert. CloudWatch

- Um Tags für eine CloudWatch Ressource festlegen oder aktualisieren zu können, müssen Sie mit einem Konto angemeldet sein, das über die `cloudwatch:TagResource` entsprechende Berechtigung verfügt. Um beispielsweise einen Alarm zu erstellen und ihm Tags zuzuweisen, müssen Sie zusätzlich zur entsprechenden `cloudwatch:TagResource` Berechtigung auch über

die `the cloudwatch:PutMetricAlarm` entsprechende Berechtigung verfügen. Wir empfehlen Ihnen, sicherzustellen, dass jeder in Ihrer Organisation, der CloudWatch Ressourcen erstellt oder aktualisiert, über die `cloudwatch:TagResource` entsprechende Berechtigung verfügt.

- Tags können für die auf Tags basierende Autorisierungssteuerung verwendet werden. Beispielsweise können IAM-Benutzer- oder -Rollenberechtigungen Bedingungen enthalten, mit denen CloudWatch Aufrufe auf bestimmte Ressourcen auf der Grundlage ihrer Tags beschränkt werden. Beachten Sie jedoch Folgendes
  - Tags, deren Namen mit beginnen, `aws:` können nicht für die tagbasierte Autorisierungssteuerung verwendet werden.
  - Zusammengesetzte Alarme unterstützen keine auf Tags basierende Autorisierungssteuerung.

Application Signals

⚠ Application Signals befindet sich in der Vorschauversion. Wenn Sie Feedback zu dieser Funktion haben, können Sie uns unter app-signals-feedback@amazon .com kontaktieren.

Verwenden Sie CloudWatch Application Signals, um Ihre Anwendungen automatisch zu AWS aktivieren, sodass Sie den aktuellen Zustand Ihrer Anwendungen überwachen und die langfristige Anwendungsleistung anhand Ihrer Geschäftsziele verfolgen können. Application Signals bietet Ihnen einen einheitlichen, anwendungsorientierten Überblick über Ihre Anwendungen, Services und Abhängigkeiten und unterstützt Sie bei der Überwachung und Diagnose des Zustands Ihrer Anwendungen.

- Aktivieren Sie Application Signals, um automatisch Metriken und Traces aus Ihren Anwendungen zu erfassen und wichtige Metriken wie Anrufvolumen, Verfügbarkeit, Latenz, Störungen und Fehler anzuzeigen. Schnell den aktuellen Betriebsstatus sehen und untersuchen und feststellen, ob Ihre Anwendungen ihre längerfristigen Leistungsziele erreichen, ohne benutzerdefinierten Code schreiben oder Dashboards erstellen zu müssen.
- Erstellen und überwachen Sie mit Application Signals Servicelevel-Ziele (SLOs). Erstellen und verfolgen Sie ganz einfach den Status von SLOs in Bezug auf CloudWatch Metriken, einschließlich der neuen Standardanwendungsmetriken, die Application Signals erfasst. Sehen und verfolgen Sie den Status des Servicelevel-Indikator (SLI) Ihrer Anwendungsservices in einer Serviceliste und einer Topologieübersicht. Erstellen Sie Alarme, um Ihre SLOs zu verfolgen, und verfolgen Sie die neuen Standard-Anwendungsmetriken, die Application Signals erfasst.
- Sehen Sie sich eine Karte Ihrer Anwendungstopologie an, die Application Signals automatisch erkennt und die Ihnen eine visuelle Darstellung Ihrer Anwendungen, Abhängigkeiten und deren Konnektivität bietet.
- Application Signals funktioniert mit CloudWatch RUM, CloudWatchSynthetics Canaries und Amazon EC2 Auto Scaling zur Anzeige Ihrer Kundenseiten, Synthetics Canaries und Anwendungsnamen in Dashboards und Maps. AWS Service Catalog AppRegistry

Application Signals für die tägliche Anwendungsüberwachung verwenden

Verwenden Sie Application Signals innerhalb der CloudWatch Konsole als Teil der täglichen Anwendungsüberwachung:

1. Wenn Sie Servicelevel-Ziele (SLOs) für Ihre Services erstellt haben, beginnen Sie mit der Seite Servicelevel-Ziele (SLO). Auf diese Weise erhalten Sie sofort einen Überblick über den Zustand Ihrer wichtigsten Services und Vorgänge. Wählen Sie den Service- oder Vorgangsnamen für ein SLO, um die Seite mit den Service-Details zu öffnen und detaillierte Service-Informationen zur Behebung von Problemen einzusehen.
2. Öffnen Sie die Services-Seite, um eine Zusammenfassung all Ihrer Services anzuzeigen und schnell die Services mit der höchsten Fehlerrate oder Latenz zu finden. Wenn Sie SLOs erstellt haben, sehen Sie in der Services-Tabelle nach, welche Services fehlerhafte Servicelevel-Indikatoren (SLIs) aufweisen. Wenn sich ein bestimmter Service in einem fehlerhaften Zustand befindet, wählen Sie den Service aus, um die Service-Detailseite zu öffnen und Service-Vorgänge, Abhängigkeiten, Synthetics-Canarys und Client-Anfragen zu sehen. Wählen Sie einen Punkt in einem Diagramm aus, um korrelierte Traces anzuzeigen, sodass Sie Betriebsprobleme beheben und deren Grundursache ermitteln können.
3. Wenn neue Services bereitgestellt wurden oder sich die Abhängigkeiten geändert haben, öffnen Sie die Service-Karte, um Ihre Anwendungstopologie zu überprüfen. Sehen Sie sich eine Karte Ihrer Anwendungen an, die die Beziehung zwischen Clients, Synthetics-Canarys, Services und Abhängigkeiten zeigt. Sehen Sie schnell den SLI-Zustand, zeigen Sie wichtige Metriken wie Aufrufvolumen, Fehlerrate und Latenz an und lassen Sie sich auf der Seite mit den Service-Details detailliertere Informationen anzeigen.

Für die Verwendung von Application Signals fallen Gebühren an. Informationen zur CloudWatch Preisgestaltung finden Sie unter CloudWatch Amazon-Preise.

> **Note**
>
> Es ist nicht erforderlich, Application Signals zu aktivieren, um CloudWatch Synthetics, CloudWatch RUM oder CloudWatch Evidently zu verwenden. Synthetics und CloudWatch RUM arbeiten jedoch mit Application Signals zusammen, um Vorteile zu bieten, wenn Sie diese Funktionen zusammen verwenden.

Unterstützte Sprachen und Architekturen

Derzeit unterstützt Application Signals Java- und Python-Anwendungen.

Application Signals wird auf Amazon EKS, Amazon ECS und Amazon EC2 unterstützt und getestet. Auf Amazon-EKS-Clustern erkennt es automatisch die Namen Ihrer Services und Cluster. Auf

anderen Architekturen müssen Sie die Namen der Services und Umgebungen angeben, wenn Sie diese Services für Application Signals aktivieren.

Die Anweisungen zur Aktivierung von Application Signals auf Amazon EC2 sollten auf jeder Architektur funktionieren, die den CloudWatch Agenten und die AWS Distribution für unterstützt. OpenTelemetry Die Anweisungen wurden jedoch nicht auf anderen Architekturen als Amazon ECS und Amazon EC2 getestet.

Unterstützte Regionen

Für diese Vorschauversion wird Application Signals in den folgenden Regionen unterstützt.

- USA Ost (Nord-Virginia)
- USA Ost (Ohio)
- USA West (Oregon)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)
- Europa (Irland)

Vorscheuversion des -SDK

Eine Vorschauversion des SDK steht zum Download zur Verfügung.

> **Warning**
>
> API-Vorgänge und -Parameter können sich ändern, bevor Application Signals allgemein verfügbar ist. Bei diesen Änderungen handelt es sich möglicherweise um grundlegende Änderungen. Verwenden Sie die Vorschauversion des SDK nicht für Produktionszwecke.

Um das Preview-SDK zu installieren, installieren oder aktualisieren Sie zunächst die neueste Version von AWS CLI Version 2. Weitere Informationen finden Sie unter Die neueste Version der AWS CLI installieren oder aktualisieren.

Verwenden Sie dann die folgenden Befehle, um die SDK-ZIP-Datei aus dem Amazon-S3-Bucket herunterzuladen und extrahieren Sie dann ihren Inhalt. Jede SDK-ZIP-Datei enthält SDK-Anweisungen und API-Dokumentation.

> **Note**
>
> Das SDK wird in mehreren Programmiersprachen bereitgestellt, sodass Sie Application Signals APIs mit jeder dieser Programmiersprachen verwenden können. Die automatische Instrumentierung Ihrer Anwendung zum Senden von Daten an Application Signals wird jedoch nur für Java- und Python-Anwendungen unterstützt.

- Java V2 SDK: `aws s3 cp s3://application-signals-preview-sdk/awsJavaSdkV2.zip ./`
- JavaScript SDK V3: `aws s3 cp s3://application-signals-preview-sdk/jsSdkV3.zip ./`
- JavaScript SDK V2: `aws s3 cp s3://application-signals-preview-sdk/jsSdkV2.zip ./`
- Python SDK: `aws s3 cp s3://application-signals-preview-sdk/pythonSdk.zip ./`
- Kotlin SDK: `aws s3 cp s3://application-signals-preview-sdk/kotlin.zip ./`
- Android SDK: `aws s3 cp s3://application-signals-preview-sdk/andriod.zip ./`
- C++ SDK: `aws s3 cp s3://application-signals-preview-sdk/awsCppSdk.zip ./`
- PHP SDK: `aws s3 cp s3://application-signals-preview-sdk/awsSdkPhp.zip ./`
- Ruby SDK: `aws s3 cp s3://application-signals-preview-sdk/awsSdkRuby.zip ./`
- Go V2 SDK: `aws s3 cp s3://application-signals-preview-sdk/awsSdkGoV2.zip ./`
- Go V1 SDK: `aws s3 cp s3://application-signals-preview-sdk/go.zip ./`
- iOS SDK: `aws s3 cp s3://application-signals-preview-sdk/iOS.zip ./`

Themen

- Führen Sie Produkteinführungen und A/B-Experimente mit CloudWatch Evidently durch
- Verwenden Sie CloudWatch RUM

Erforderliche Berechtigungen für Application Signals

⚠ Application Signals befindet sich in der Vorschauversion für Amazon CloudWatch und kann sich ändern.

In diesem Abschnitt werden die Berechtigungen erläutert, die Sie für die Aktivierung, Verwaltung und den Betrieb von Application Signals benötigen.

Berechtigungen zur Aktivierung und Verwaltung von Application Signals

Um Application Signals verwalten zu können, müssen Sie mit den folgenden Berechtigungen angemeldet sein:

```
{
 "Version": "2012-10-17",
 "Statement": [
    {
        "Sid": "CloudWatchApplicationSignalsFullAccessPermissions",
        "Effect": "Allow",
        "Action": "application-signals:*",
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsAlarmsPermissions",
        "Effect": "Allow",
        "Action": [
            "cloudwatch:DescribeAlarms"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsMetricsPermissions",
        "Effect": "Allow",
        "Action": [
            "cloudwatch:GetMetricData",
            "cloudwatch:ListMetrics"
```

```
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsLogGroupPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:StartQuery",
            "logs:DescribeMetricFilters"
        ],
        "Resource": "arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsLogsPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:GetQueryResults",
            "logs:StopQuery"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsSyntheticsPermissions",
        "Effect": "Allow",
        "Action": [
            "synthetics:DescribeCanaries",
            "synthetics:DescribeCanariesLastRun",
            "synthetics:GetCanaryRuns"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsRumPermissions",
        "Effect": "Allow",
        "Action": [
            "rum:BatchCreateRumMetricDefinitions",
            "rum:BatchDeleteRumMetricDefinitions",
            "rum:BatchGetRumMetricDefinitions",
            "rum:GetAppMonitor",
            "rum:GetAppMonitorData",
            "rum:ListAppMonitors",
            "rum:PutRumMetricsDestination",
            "rum:UpdateRumMetricDefinition"
        ],
```

```
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsXrayPermissions",
        "Effect": "Allow",
        "Action": [
            "xray:GetTraceSummaries"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsPutMetricAlarmPermissions",
        "Effect": "Allow",
        "Action": "cloudwatch:PutMetricAlarm",
        "Resource": [
            "arn:aws:cloudwatch:*:*:alarm:SLO-AttainmentGoalAlarm-*",
            "arn:aws:cloudwatch:*:*:alarm:SLO-WarningAlarm-*",
            "arn:aws:cloudwatch:*:*:alarm:SLI-HealthAlarm-*"
        ]
    },
    {
        "Sid": "CloudWatchApplicationSignalsCreateServiceLinkedRolePermissions",
        "Effect": "Allow",
        "Action": "iam:CreateServiceLinkedRole",
        "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals",
        "Condition": {
            "StringLike": {
                "iam:AWSServiceName": "application-signals.cloudwatch.amazonaws.com"
            }
        }
    },
    {
        "Sid": "CloudWatchApplicationSignalsGetRolePermissions",
        "Effect": "Allow",
        "Action": "iam:GetRole",
        "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals"
    },
    {
        "Sid": "CloudWatchApplicationSignalsSnsWritePermissions",
        "Effect": "Allow",
        "Action": [
            "sns:CreateTopic",
```

```
            "sns:Subscribe"
        ],
        "Resource": "arn:aws:sns:*:*:cloudwatch-application-signals-*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsSnsReadPermissions",
        "Effect": "Allow",
        "Action": "sns:ListTopics",
        "Resource": "*"
    }
 ]
}
```

Informationen zur Aktivierung von Application Signals auf Amazon EC2 oder benutzerdefinierten Architekturen finden Sie unter Aktivieren von Application Signals auf anderen Plattformen mit einer benutzerdefinierten Konfiguration. Kubernetes Um Application Signals auf Amazon EKS mithilfe des Amazon CloudWatch Observability EKS-Add-ons zu aktivieren und zu verwalten, benötigen Sie die folgenden Berechtigungen.

> **Important**
>
> Zu diesen Berechtigungen gehören `iam:PassRole` mit `Resource "*”` und `eks:CreateAddon` mit `Resource “*”`. Dabei handelt es sich um leistungsstarke Berechtigungen, und Sie sollten bei der Erteilung dieser Berechtigungen Vorsicht walten lassen.

```
{
"Version": "2012-10-17",
    "Statement": [
        {
"Sid": "CloudWatchApplicationSignalsEksAddonManagementPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:AccessKubernetesApi",
                "eks:CreateAddon",
                "eks:DescribeAddon",
                "eks:DescribeAddonConfiguration",
                "eks:DescribeAddonVersions",
                "eks:DescribeCluster",
                "eks:DescribeUpdate",
```

```
                "eks:ListAddons",
                "eks:ListClusters",
                "eks:ListUpdates",
                "iam:ListRoles",
                "iam:PassRole"
            ],
            "Resource": "*"
        },
        {
"Sid":
 "CloudWatchApplicationSignalsEksCloudWatchObservabilityAddonManagementPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:DeleteAddon",
                "eks:UpdateAddon"
            ],
            "Resource": "arn:aws:eks:*:*:addon/*/amazon-cloudwatch-observability/*"
        }
    ]
}
```

Das Application Signals-Dashboard zeigt die AWS Service Catalog AppRegistry Anwendungen, mit denen Ihre SLOs verknüpft sind. Um diese Anwendungen auf den SLO-Seiten zu sehen, benötigen Sie die folgenden Berechtigungen:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsTaggingReadPermissions",
            "Effect": "Allow",
            "Action": "tag:GetResources",
            "Resource": "*"
        }
    ]
}
```

Betrieb von Application Signals

Dienstanbieter, die Application Signals zur Überwachung von Diensten und SLOs verwenden, müssen mit einem Konto angemeldet sein, das über die folgenden Leseberechtigungen verfügt:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsReadOnlyAccessPermissions",
            "Effect": "Allow",
            "Action": [
                "application-signals:BatchGet*",
                "application-signals:Get*",
                "application-signals:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsGetRolePermissions",
            "Effect": "Allow",
            "Action": "iam:GetRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals"
        },
        {
        "Sid": "CloudWatchApplicationSignalsLogGroupPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:StartQuery",
            "logs:DescribeMetricFilters"
        ],
        "Resource": "arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsLogsPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:GetQueryResults",
            "logs:StopQuery"
        ],
        "Resource": "*"
    },
        {
            "Sid": "CloudWatchApplicationSignalsAlarmsReadPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms"
```

```
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsMetricsReadPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:ListMetrics"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsSyntheticsReadPermissions",
            "Effect": "Allow",
            "Action": [
                "synthetics:DescribeCanaries",
                "synthetics:DescribeCanariesLastRun",
                "synthetics:GetCanaryRuns"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsRumReadPermissions",
            "Effect": "Allow",
            "Action": [
                "rum:BatchGetRumMetricDefinitions",
                "rum:GetAppMonitor",
                "rum:GetAppMonitorData",
                "rum:ListAppMonitors"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsXrayReadPermissions",
            "Effect": "Allow",
            "Action": [
                "xray:GetTraceSummaries"
            ],
            "Resource": "*"
        }
    ]
}
```

Um im Application Signals-Dashboard zu sehen, welchen AWS Service Catalog AppRegistry Anwendungen Ihre SLOs zugeordnet sind, benötigen Sie die folgenden Berechtigungen:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsTaggingReadPermissions",
            "Effect": "Allow",
            "Action": "tag:GetResources",
            "Resource": "*"
        }
    ]
}
```

Um zu überprüfen, ob Application Signals auf Amazon EKS mit dem Amazon CloudWatch Observability EKS-Add-on aktiviert ist, benötigen Sie die folgenden Berechtigungen:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsEksReadPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:ListAddons",
                "eks:ListClusters"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsEksDescribeAddonReadPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:DescribeAddon"
            ],
            "Resource": "arn:aws:eks:*:*:addon/*/amazon-cloudwatch-observability/*"
        }
    ]
}
```

Application Signals aktivieren

⚠ Application Signals befindet sich in der Vorschauversion. Wenn Sie Feedback zu dieser Funktion haben, können Sie uns unter app-signals-feedback@amazon .com kontaktieren.

In den Themen in diesem Abschnitt wird erklärt, wie Sie CloudWatch Application Signals in Ihrer Umgebung aktivieren. Application Signals wird auf Amazon-EKS-Clustern mit einem Einrichtungs-Workflow über die Konsole unterstützt. Das Feature wird auch auf anderen Plattformen, einschließlich Amazon EC2, mit einem benutzerdefinierten Einrichtungsprozess unterstützt.

Themen

Application Signals, unterstützte Systeme

⚠ Application Signals befindet sich in der Vorschauversion. Wenn Sie Feedback zu dieser Funktion haben, können Sie uns unter app-signals-feedback@amazon .com kontaktieren.

Application Signals wird auf Amazon EKS, Amazon ECS und Amazon EC2 unterstützt und getestet. Die Anweisungen zur Aktivierung von Application Signals auf Amazon EC2 sollten auf jeder Plattform funktionieren, die den CloudWatch Agenten und die AWS Distribution für unterstützt OpenTelemetry, aber die Anweisungen wurden nicht auf anderen Plattformen getestet.

Java-Kompatibilität

Application Signals unterstützt Java-Anwendungen und unterstützt dieselben Java-Bibliotheken und Frameworks wie AWS Distro for. OpenTelemetry Weitere Informationen finden Sie unter Unterstützte Bibliotheken, Frameworks, Anwendungsserver und JVMs.

Die JVM-Versionen 8, 11 und 17 werden unterstützt.

Python-Kompatibilität

Application Signals unterstützt dieselben Bibliotheken und Frameworks wie die AWS Distribution for OpenTelemetry . Weitere Informationen finden Sie unter Unterstützte Pakete unter opentelemetry-python-contrib.

Python-Versionen 3.8 und höher werden unterstützt.

Bevor Sie Application Signals für Ihre Python-Anwendungen aktivieren, sollten Sie die folgenden Überlegungen beachten.

- In einigen containerisierten Anwendungen kann eine fehlende `PYTHONPATH` Umgebungsvariable manchmal dazu führen, dass die Anwendung nicht gestartet werden kann. Um dieses Problem zu beheben, stellen Sie sicher, dass Sie die `PYTHONPATH` Umgebungsvariable auf den Speicherort des Arbeitsverzeichnisses Ihrer Anwendung setzen. Dies ist auf ein bekanntes Problem mit der OpenTelemetry automatischen Instrumentierung zurückzuführen. Weitere Informationen zu diesem Problem finden Sie unter Python-Autoinstrumentation-Einstellung von PYTHONPATH ist nicht kompatibel.
- Für Django-Anwendungen sind zusätzliche Konfigurationen erforderlich, die in der OpenTelemetry Python-Dokumentation beschrieben werden.
  - Verwenden Sie das `--noreload` Flag, um ein automatisches Neuladen zu verhindern.
  - Setzen Sie die `DJANGO_SETTINGS_MODULE` Umgebungsvariable auf den Speicherort der Datei Ihrer Django-Anwendung. `settings.py` Dadurch wird sichergestellt, dass OpenTelemetry Sie korrekt auf Ihre Django-Einstellungen zugreifen und diese integrieren können.

OpenTelemetry Überlegungen zur Kompatibilität

⚠ Application Signals befindet sich in der Vorschauversion. Wenn Sie Feedback zu dieser Funktion haben, können Sie uns unter app-signals-feedback@amazon .com kontaktieren.

Um Ihre Anwendungen mit CloudWatch Application Signals zu integrieren, empfehlen wir Ihnen, zuvor alle vorhandenen Lösungen zur Überwachung der Anwendungsleistung vollständig aus Ihrer Anwendung zu entfernen. Dies beinhaltet das Entfernen von jeglichem Instrumentierungscode und Konfigurationen.

Auch wenn Application Signals OpenTelemetry Instrumentierung verwendet, kann nicht garantiert werden, dass diese mit Ihrer vorhandenen OpenTelemetry Instrumentierung oder Konfiguration kompatibel ist. Im besten Fall können Sie möglicherweise einige Ihrer OpenTelemetry Funktionen beibehalten, z. B. benutzerdefinierte Metriken. Lesen Sie jedoch unbedingt die folgenden Abschnitte mit Informationen.

Überlegungen, falls Sie bereits Folgendes verwenden OpenTelemetry

Wenn Sie es bereits OpenTelemetry zusammen mit Ihrer Anwendung verwenden, enthält der Rest dieses Abschnitts wichtige Informationen, um die Kompatibilität mit Application Signals zu erreichen.

- Bevor Sie Ihre Anwendung für Application Signals aktivieren, müssen Sie die Injektion aller anderen Autoinstrumentierungs-Agenten, die auf OpenTelemetry Ihrer Anwendung basieren, entfernen. Dies hilft, Konfigurationskonflikte zu vermeiden. Sie können weiterhin die manuelle Instrumentierung mithilfe kompatibler OpenTelemetry APIs zusammen mit Application Signals verwenden.
- Wenn Sie manuelle Instrumentierung verwenden, um benutzerdefinierte Spans oder Metriken aus Ihrer Anwendung zu generieren, kann die Aktivierung von Application Signals je nach Komplexität der Instrumentierung dazu führen, dass sie keine Daten mehr generieren oder ein anderes unerwünschtes Verhalten zeigen. Möglicherweise können Sie einige der verfügbaren Konfigurationen OpenTelemetry (mit Ausnahme der in der Tabelle weiter unten in diesem Abschnitt genannten) verwenden, um das gewünschte Verhalten Ihrer vorhandenen Metriken oder Spans beizubehalten. Weitere Informationen zu diesen Konfigurationen finden Sie in der OpenTelemetry Dokumentation unter SDK-Konfiguration.

  Wenn Sie beispielsweise die `OTEL_EXPORTER_OTLP_METRICS_ENDPOINT` Konfiguration und eine selbstverwaltete OpenTelemetry Collector-Instanz verwenden, können Sie Ihre benutzerdefinierten Messwerte möglicherweise weiterhin an das gewünschte Ziel senden.
- Einige Umgebungsvariablen oder Systemeigenschaften dürfen nicht mit Application Signals verwendet werden, während Sie andere verwenden können, sofern Sie die Anweisungen in der Tabelle befolgen. Einzelheiten sind der folgenden Tabelle zu entnehmen.

Umgebungsvariable	Empfehlung mit Application Signals
Allgemeine Umgebungsvariablen	
`OTEL_SDK_DISABLED`	Darf nicht auf `true` festgelegt sein.

Umgebungsvariable	Empfehlung mit Application Signals
`OTEL_TRACES_EXPORTER`	Muss auf `otlp` festgelegt sein.
`OTEL_EXPORTER_OTLP_ENDPOINT`	Darf nicht verwendet werden.
`OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`	Darf nicht verwendet werden.
`OTEL_ATTRIBUTE_COUNT_LIMIT`	Falls gesetzt, muss es hoch genug gesetzt werden, um etwa 10 weitere Span-Attribute aufzunehmen, die von CloudWatch Application Signals hinzugefügt werden.
`OTEL_PROPAGATORS`	Falls festgelegt, muss es `xray` für die Endverfolgung enthalten.
`OTEL_TRACES_SAMPLER`	Falls festgelegt, muss `xray` sein, um das zentralisierte X-Ray Sampling zu verwenden. Um das lokale Sampling zu verwenden, stellen Sie diesen Wert auf `parentbased_traceidratio` ein und geben Sie die Sampling-Rate in `OTEL_TRACES_SAMPLER_ARG` ein.
`OTEL_TRACES_SAMPLER_ARG`	Wenn Sie die Standardeinstellung des zentralisierten X-Ray Trace Sample verwenden, darf diese Variable nicht verwendet werden. Wenn Sie stattdessen das lokale Sampling verwenden, legen Sie die Sampling-Rate in dieser Variable fest. Zum Beispiel `0.05` für eine Sampling-Rate von 5 %.
Java-spezifische Umgebungsvariablen	
`OTEL_JAVA_ENABLED_RESOURCE_PROVIDERS`	Falls gesetzt, müssen sie AWS Ressourcendetektoren enthalten.

Umgebungsvariable	Empfehlung mit Application Signals
Python-spezifische Umgebungsvariablen	
`OTEL_PYTHON_CONFIGURATOR`	Falls verwendet, muss sie auf gesetzt werden `aws_configurator`
`OTEL_PYTHON_DISTRO`	Falls verwendet, muss auf gesetzt werden `aws_distro`

Application Signals auf Amazon-EKS-Clustern aktivieren

⚠ Application Signals befindet sich in der Vorschauversion. Wenn Sie Feedback zu dieser Funktion haben, können Sie uns unter app-signals-feedback@amazon .com kontaktieren.

CloudWatch Application Signals wird für Java- und Python-Anwendungen unterstützt, die in Amazon EKS-Clustern ausgeführt werden. Sie haben zwei Möglichkeiten, Application Signals für Anwendungen in einem Amazon-EKS-Cluster zu aktivieren:

- Um Application Signals für Ihre Anwendungen auf einem vorhandenen Amazon-EKS-Cluster zu aktivieren, verwenden Sie die Schritte unter Application Signals auf einem Amazon-EKS-Cluster mit Ihren Services aktivieren.
- Verwenden Sie die Anweisungen unter Application Signals auf einem neuen Amazon-EKS-Cluster mit einer Beispiel-App aktivieren, um Application Signals in einer Umgebung außerhalb der Produktionsumgebung mit einer Beispielanwendung auszuprobieren. Dieser Workflow verwendet Skripts, die von AWS bereitgestellt werden, um einen neuen Amazon-EKS-Cluster zu erstellen und eine Beispielanwendung zu installieren, die für Application Signals aktiviert ist. Auf diese Weise können Sie die end-to-end Funktionalität von Application Signals sehen und testen.

Themen

Application Signals auf einem Amazon-EKS-Cluster mit Ihren Services aktivieren

⚠ Application Signals befindet sich in der Vorschauversion. Wenn Sie Feedback zu dieser Funktion haben, können Sie uns unter app-signals-feedback@amazon .com kontaktieren.

Verwenden Sie die Anweisungen in diesem Abschnitt, um CloudWatch Application Signals für Ihre Anwendungen auf einem vorhandenen Amazon EKS-Cluster zu aktivieren.

⚠ Important

Wenn Sie bereits eine Anwendung verwenden OpenTelemetry , die Sie für Application Signals aktivieren möchten, finden Sie weitere Informationen, OpenTelemetry Überlegungen zur Kompatibilität bevor Sie Application Signals aktivieren.

So aktivieren Sie Application Signals für Ihre Anwendungen auf einem vorhandenen Amazon-EKS-Cluster

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Services.
3. Wenn Sie Application Signals in diesem Konto noch nicht aktiviert haben, müssen Sie Application Signals die Berechtigungen gewähren, die es benötigt, um Ihre Services zu erkennen. Gehen Sie dazu wie folgt vor. Dies muss nur einmal für Ihr Konto durchgeführt werden.

   a. Wählen Sie Mit der Entdeckung Ihrer Services beginnen.

   b. Aktivieren Sie das Kontrollkästchen und wählen Sie Mit der Entdeckung von Services beginnen.

      Wenn Sie diesen Schritt zum ersten Mal in Ihrem Konto ausführen, wird die AWSServiceRoleForCloudWatchApplicationSignalsdienstverknüpfte Rolle erstellt. Diese Rolle gewährt Application Signals die folgenden Berechtigungen:

      - `xray:GetServiceGraph`
      - `logs:StartQuery`
      - `logs:GetQueryResults`

- `cloudwatch:GetMetricData`
- `cloudwatch:ListMetrics`
- `tag:GetResources`

Weitere Informationen über diese Rolle finden Sie unter Dienstbezogene Rollenberechtigungen für Anwendungssignale CloudWatch .

4. Wählen Sie Application Signals aktivieren.
5. Wählen Sie für Plattform angeben die Option EKS.
6. Wählen Sie für Einen EKS-Cluster auswählen den Cluster aus, in dem Sie Application Signals aktivieren möchten.
7. Wenn für diesen Cluster das Amazon CloudWatch Observability EKS-Add-on noch nicht aktiviert ist, werden Sie aufgefordert, es zu aktivieren. In diesem Fall, gehen Sie wie folgt vor:

   a. Wählen Sie CloudWatch Observability EKS-Add-on hinzufügen. Die Amazon-EKS-Konsole wird angezeigt.

   b. Aktivieren Sie das Kontrollkästchen für Amazon CloudWatch Observability und wählen Sie Weiter.

      Das CloudWatch Observability EKS-Add-on ermöglicht sowohl Application Signals als auch CloudWatch Container Insights mit verbesserter Observability für Amazon EKS. Weitere Informationen zu Container Insights finden Sie unter Container Insights.

   c. Wählen Sie die neueste Version des zu installierenden Add-Ons.

   d. Wählen Sie eine IAM-Rolle aus, die für das Add-On verwendet werden soll. Wenn Sie Von Knoten erben wählen, fügen Sie der IAM-Rolle, die von Ihren Worker-Knoten verwendet wird, die richtigen Berechtigungen hinzu. *my-worker-node-role*Ersetzen Sie es durch die IAM-Rolle, die von Ihren Kubernetes-Worker-Knoten verwendet wird.

      ```
      aws iam attach-role-policy \
      --role-name my-worker-node-role \
      --policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy \
      --policy-arn arn:aws:iam::aws:policy/AWSXRayWriteOnlyAccess
      ```

   e. Informationen zum Erstellen einer Servicerolle für die Verwendung des Add-Ons finden Sie unter Installieren Sie den CloudWatch Agenten mithilfe des Amazon CloudWatch Observability EKS-Add-ons.

f. Wählen Sie Weiter, bestätigen Sie die Informationen auf dem Bildschirm und wählen Sie Erstellen.

g. Wählen Sie im nächsten Bildschirm Enable CloudWatch Application Signals aus, um zur CloudWatch Konsole zurückzukehren und den Vorgang abzuschließen.

8. Es gibt zwei Optionen, um Ihre Anwendungen für Application Signals zu aktivieren. Aus Konsistenzgründen empfehlen wir, dass Sie eine Option pro Cluster wählen.

- Die Konsolenoption ist einfacher. Wenn Sie diese Methode verwenden, werden Ihre Pods sofort neu gestartet.
- Mit der Methode Annotate Manifest File haben Sie mehr Kontrolle darüber, wann Ihre Pods neu gestartet werden. Außerdem können Sie Ihre Überwachung dezentraler verwalten, falls Sie sie nicht zentralisieren möchten.

Console

Die Konsolenoption verwendet die erweiterte Konfiguration des Amazon CloudWatch Observability EKS-Add-ons, um Application Signals für Ihre Services einzurichten. Weitere Informationen über das Add-on finden Sie unter(Optional) Zusätzliche Konfiguration.

Wenn Sie keine Liste mit Workloads und Namespaces sehen, stellen Sie sicher, dass Sie über die entsprechenden Berechtigungen verfügen, um sie für diesen Cluster anzuzeigen. Weitere Informationen finden Sie unter Erforderliche Berechtigungen.

Sie können einzelne Workloads oder ganze Namespaces überwachen.

Um einen einzelnen Workload zu überwachen:

1. Aktivieren Sie das Kontrollkästchen neben dem Workload, den Sie überwachen möchten.
2. Wählen Sie die Sprache des Workloads aus. Stellen Sie bei Python-Anwendungen sicher, dass Ihre Anwendung die erforderlichen Voraussetzungen erfüllt, bevor Sie fortfahren. Weitere Informationen finden Sie unter Die Python-Anwendung startet nicht, nachdem Application Signals aktiviert wurde.
3. Wählen Sie Erledigt aus. Das Amazon CloudWatch Observability EKS-Add-on fügt sofort AWS DISTRO for OpenTelemetry Autoinstrumentation (ADOT) -SDKs in Ihre Pods ein und löst Pod-Neustarts aus, um die Erfassung von Anwendungsmetriken und Traces zu ermöglichen.

Um einen ganzen Namespace zu überwachen:

1. Aktivieren Sie das Kontrollkästchen neben dem Namespace, den Sie überwachen möchten.
2. Wählen Sie die Sprache des Workloads aus. Dies gilt für alle Workloads in diesem Namespace, unabhängig davon, ob sie derzeit bereitgestellt werden oder in future bereitgestellt werden. Stellen Sie bei Python-Anwendungen sicher, dass Ihre Anwendung die erforderlichen Voraussetzungen erfüllt, bevor Sie fortfahren. Weitere Informationen finden Sie unter Die Python-Anwendung startet nicht, nachdem Application Signals aktiviert wurde.
3. Wählen Sie Erledigt aus. Das Amazon CloudWatch Observability EKS-Add-on fügt sofort AWS DISTRO for OpenTelemetry Autoinstrumentation (ADOT) -SDKs in Ihre Pods ein und löst Pod-Neustarts aus, um die Erfassung von Anwendungsmetriken und Traces zu ermöglichen.

Um Application Signals in einem anderen Amazon-EKS-Cluster zu aktivieren, wählen Sie auf dem Services-Bildschirm die Option Application Signals aktivieren.

Annotate manifest file

In der CloudWatch Konsole wird im Abschnitt Monitor Services erklärt, dass Sie einer Manifest-YAML im Cluster eine Anmerkung hinzufügen müssen. Durch das Hinzufügen dieser Anmerkung wird die Anwendung automatisch so instrumentiert, dass sie Metriken, Traces und Protokolle an Application Signals sendet.

Sie haben zwei Möglichkeiten für die Anmerkung:

- Workload kommentieren instrumentiert automatisch einen einzelnen Workload im Cluster.
- Namespace kommentieren instrumentiert automatisch alle Workloads, die im ausgewählten Namespace bereitgestellt werden.

Wählen Sie eine dieser Optionen und folgen Sie den entsprechenden Schritten:

- Um einen einzelnen Workload mit Anmerkungen zu versehen:
  1. Wählen Sie Workload kommentieren.

2. Fügen Sie eine der folgenden Zeilen in den `PodTemplate` Abschnitt der Workload-Manifestdatei ein.
   - Für Java-Workloads: `annotations: instrumentation.opentelemetry.io/inject-java: "true"`
   - Für Python-Workloads: `annotations: instrumentation.opentelemetry.io/inject-python: "true"`

     Für Python-Anwendungen sind zusätzliche Konfigurationen erforderlich. Weitere Informationen finden Sie unter Die Python-Anwendung startet nicht, nachdem Application Signals aktiviert wurde.
3. Geben Sie in Ihrem Terminal `kubectl apply -f` *`your_deployment_yaml`* ein, um die Änderung zu übernehmen.

- Um alle Workloads in einem Namespace mit Anmerkungen zu versehen:
  1. Wählen Sie Namespace kommentieren.
  2. Fügen Sie eine der folgenden Zeilen in den Metadatenbereich der Namespace-Manifestdatei ein. Wenn der Namespace sowohl Java- als auch Python-Workloads umfasst, fügen Sie beide Zeilen in die Namespace-Manifestdatei ein.
     - Wenn es Java-Workloads im Namespace gibt: `annotations: instrumentation.opentelemetry.io/inject-java: "true"`
     - Wenn es Python-Workloads im Namespace gibt: `annotations: instrumentation.opentelemetry.io/inject-python: "true"`

       Für Python-Anwendungen sind zusätzliche Konfigurationen erforderlich. Weitere Informationen finden Sie unter Die Python-Anwendung startet nicht, nachdem Application Signals aktiviert wurde.
  3. Geben Sie in Ihrem Terminal `kubectl apply -f` *`your_namespace_yaml`* ein, um die Änderung zu übernehmen.
  4. Geben Sie in Ihrem Terminal einen Befehl ein, um alle Pods im Namespace neu zu starten. Ein Beispielbefehl zum Neustarten von Bereitstellungs-Workloads ist `kubectl rollout restart deployment -n` *`namespace_name`*

9. Wählen Sie Nach Abschluss Services anzeigen. Dadurch gelangen Sie zur Services-Ansicht von Application Signals, in der Sie die Daten sehen können, die Application Signals sammelt. Es kann einige Minuten dauern, bis Daten angezeigt werden.

Um Application Signals in einem anderen Amazon-EKS-Cluster zu aktivieren, wählen Sie auf dem Services-Bildschirm die Option Application Signals aktivieren.

Weitere Informationen über die Services-Ansicht finden Sie unter Den Betriebsstatus Ihrer Anwendungen mit Application Signals überwachen.

> **Note**
>
> Wir haben einige Überlegungen identifiziert, die Sie bei der Aktivierung von Python-Anwendungen für Application Signals berücksichtigen sollten. Weitere Informationen finden Sie unter Die Python-Anwendung startet nicht, nachdem Application Signals aktiviert wurde.

Application Signals auf einem neuen Amazon-EKS-Cluster mit einer Beispiel-App aktivieren

> **Application Signals befindet sich in der Vorschauversion. Wenn Sie Feedback zu dieser Funktion haben, können Sie uns unter app-signals-feedback@amazon .com kontaktieren.**

Um CloudWatch Application Signals in einer Beispiel-App auszuprobieren, bevor Sie Ihre eigenen Anwendungen damit instrumentieren, folgen Sie den Anweisungen in diesem Abschnitt. Diese Anweisungen verwenden Skripts, um Ihnen zu helfen, einen Amazon-EKS-Cluster zu erstellen, eine Beispielanwendung zu installieren und die Beispielanwendung so zu instrumentieren, dass sie mit Application Signals funktioniert.

Bei der Beispielanwendung handelt es sich um eine Spring-„Pet-Clinic“-Anwendung, die aus vier Microservices besteht. Diese Services werden auf Amazon EKS auf Amazon EC2 ausgeführt und nutzen Application-Signals-Aktivierungsskripten, um den Cluster mit dem Java- oder Python-Agenten für automatische Instrumentierung zu aktivieren.

Voraussetzungen

- Derzeit überwacht Application Signals nur Java- und Python-Anwendungen.

- Sie müssen das auf der Instanz AWS CLI installiert haben. Wir empfehlen AWS CLI Version 2, aber Version 1 sollte auch funktionieren. Weitere Informationen zur Installation von finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI. AWS CLI
- Die Skripts in diesem Abschnitt sind für die Ausführung in Linux- und macOS-Umgebungen vorgesehen. Für Windows-Instances empfehlen wir, dass Sie eine AWS Cloud9 Umgebung verwenden, um diese Skripts auszuführen. Weitere Informationen zu AWS Cloud9 finden Sie unter Was ist AWS Cloud9? .
- Installieren Sie eine unterstützte Version von `kubectl`. Sie müssen eine `kubectl`-Version verwenden, die maximal um eine Nebenversion von der Steuerebene Ihres Amazon-EKS-Clusters abweicht. Ein `kubectl`-Client der Version 1.26 funktioniert beispielsweise mit den Kubernetes-Cluster-Versionen 1.25, 1.26 und 1.27. Wenn Sie bereits über einen Amazon EKS-Cluster verfügen, müssen Sie möglicherweise AWS Anmeldeinformationen für konfigurieren`kubectl`. Weitere Informationen finden Sie unter Erstellen oder Aktualisieren einer `kubeconfig`-Datei für einen Amazon-EKS-Cluster.
- Installieren`eksctl`. `eksctl`verwendet das, AWS CLI um mit dem zu interagieren AWS, was bedeutet, dass es dieselben AWS Anmeldeinformationen verwendet wie der AWS CLI. Weitere Informationen finden Sie unter Installieren oder Aktualisieren von `eksctl`.
- Installieren Sie `jq`. `jq` ist erforderlich, um die Aktivierungsskripts für Application Signals auszuführen. Weitere Informationen finden Sie unter jq herunterladen.

Schritt 1: Die Skripte herunterladen

Um die Skripts für die Einrichtung von CloudWatch Application Signals mit einer Beispiel-App herunterzuladen, können Sie die komprimierte GitHub Projektdatei auf ein lokales Laufwerk herunterladen und dekomprimieren, oder Sie können das GitHub Projekt klonen.

Wenn Sie das Projekt klonen möchten, öffnen Sie ein Terminalfenster und geben Sie den folgenden Git-Befehl in ein bestimmtes Arbeitsverzeichnis ein.

```
git clone https://github.com/aws-observability/application-signals-demo.git
```

Schritt 2: Die Beispielanwendung entwickeln und bereitstellen

Folgen Sie diesen Anweisungen, um die Beispielanwendungs-Images zu erstellen und zu übertragen.

Schritt 3: Application Signals und die Beispielanwendung bereitstellen und aktivieren

Vergewissern Sie sich, dass Sie die unter Application Signals auf einem neuen Amazon-EKS-Cluster mit einer Beispiel-App aktivieren aufgeführten Anforderungen erfüllt haben, bevor Sie die folgenden Schritte ausführen.

So können Sie Application Signals und die Beispielanwendung bereitstellen und aktivieren

1. Geben Sie den folgenden Befehl in das lokale Terminal ein, in dem Sie das Onboarding-Skript entpackt haben. *new-cluster-name*Ersetzen Sie es durch den Namen, den Sie für den neuen Cluster verwenden möchten. Ersetzen Sie den *Regionsnamen* durch den Namen der AWS Region, z. B. `us-west-1`

   Dieser Befehl richtet die Beispiel-App ein, die in einem neuen Amazon-EKS-Cluster mit Application Signals aktiviert ausgeführt wird.

   ```
   # assuming the current working directory is 'onboarding'
   # this script sets up a new cluster, enables Application Signals, and deploys the
   # sample application
   cd application-signals-demo/scripts/eks/appsignals/one-step && ./setup.sh new-
   cluster-name region-name
   ```

   Die Ausführung des Einrichtungs-Skripts dauert etwa 30 Minuten und macht Folgendes:

   - Erstellt einen neuen Amazon-EKS-Cluster in der angegebenen Region.
   - Erstellt die erforderlichen IAM-Berechtigungen für Application Signals (`arn:aws:iam::aws:policy/AWSXrayWriteOnlyAccess` und`arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy`).
   - Aktiviert Application Signals, indem der CloudWatch Agent installiert und die Beispielanwendung automatisch für CloudWatch Metriken und Röntgen-Traces instrumentiert wird.
   - Stellt die PetClinic Spring-Beispielanwendung im selben Amazon EKS-Cluster bereit.
   - Erzeugt fünf CloudWatch Synthetics-Kanarienvögel mit dem Namen`pc-add-vist`,,`pc-create-owners`,`pc-visit-pet`,`pc-visit-vet`. `pc-clinic-traffic` Diese Canarys werden mit einer Frequenz von einer Minute ausgeführt, um synthetischen Datenverkehr für die Beispielanwendung zu generieren und zu demonstrieren, wie Synthetics-Canarys in Application Signals erscheinen.

- Erstellt vier Service Level Objectives (SLOs) für die PetClinic Anwendung mit den folgenden Namen:
  - Verfügbarkeit für die Suche nach einem Eigentümer
  - Latenz bei der Suche nach einem Eigentümer
  - Verfügbarkeit für die Registrierung eines Besitzers
  - Latenz bei der Registrierung eines Besitzers
- Erstellt die erforderliche IAM-Rolle mit einer benutzerdefinierten Vertrauensrichtlinie, die Application Signals die folgenden Berechtigungen gewährt:
  - `cloudwatch:PutMetricData`
  - `cloudwatch:GetMetricData`
  - `xray:GetServiceGraph`
  - `logs:StartQuery`
  - `logs:GetQueryResults`

2. (Optional) Wenn Sie den Quellcode für die PetClinic Beispielanwendung überprüfen möchten, finden Sie sie im Stammordner.

```
- application-signals-demo
  - spring-petclinic-admin-server
  - spring-petclinic-api-gateway
  - spring-petclinic-config-server
  - spring-petclinic-customers-service
  - spring-petclinic-discovery-server
  - spring-petclinic-vets-service
  - spring-petclinic-visits-service
```

3. Um die bereitgestellte PetClinic Beispielanwendung anzuzeigen, führen Sie den folgenden Befehl aus, um die URL zu finden:

```
kubectl get ingress
```

Schritt 4: Die Beispielanwendung überwachen

Nachdem Sie die Schritte im vorherigen Abschnitt ausgeführt haben, um den Amazon-EKS-Cluster zu erstellen und die Beispielanwendung bereitzustellen, können Sie Application Signals verwenden, um die Anwendung zu überwachen.

> **Note**
>
> Damit die Application-Signals-Konsole Daten aufnehmen kann, muss ein Teil des Datenverkehrs die Beispielanwendung erreichen. Im Rahmen der vorherigen Schritte wurden CloudWatch Synthetics Canaries erstellt, die Traffic zur Beispielanwendung generieren.

Überwachen des Service-Zustands

Nach der Aktivierung erkennt CloudWatch Application Signals automatisch eine Liste von Diensten und füllt sie aus, ohne dass eine zusätzliche Einrichtung erforderlich ist.

So zeigen Sie die Liste der erkannten Services an und überwachen deren Zustand

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Services.
3. Um einen Service, seine Vorgänge und seine Abhängigkeiten anzuzeigen, wählen Sie den Namen eines der Services in der Liste aus.

   Diese einheitliche, anwendungsorientierte Ansicht bietet einen vollständigen Überblick darüber, wie Benutzer mit Ihrem Service interagieren. Dies kann Ihnen helfen, Probleme zu beheben, wenn Leistungsanomalien auftreten. Vollständige Informationen zur Services-Ansicht finden Sie unter Den Betriebsstatus Ihrer Anwendungen mit Application Signals überwachen.
4. Wählen Sie die Registerkarte Service-Vorgänge, um die Standard-Anwendungsmetriken für die Vorgänge dieses Services anzuzeigen. Bei den Vorgängen handelt es sich beispielsweise um die API-Vorgänge, die der Service aufruft.

   Um dann die Diagramme für einen einzelnen Vorgang dieses Services anzuzeigen, wählen Sie den Namen des Vorgangs aus.
5. Wählen Sie die Registerkarte Abhängigkeiten, um die Abhängigkeiten Ihrer Anwendung sowie die kritischen Anwendungsmetriken für jede Abhängigkeit anzuzeigen. Zu den Abhängigkeiten gehören AWS Dienste und Dienste von Drittanbietern, die Ihre Anwendung aufruft.
6. Um korrelierte Traces auf der Seite mit den Service-Details anzuzeigen, wählen Sie einen Datenpunkt in einem der drei Diagramme über der Tabelle aus. Dadurch wird ein neuer Bereich mit gefilterten Traces aus dem Zeitraum gefüllt. Diese Traces werden auf der Grundlage des ausgewählten Diagramms sortiert und gefiltert. Wenn Sie beispielsweise das Latenzdiagramm ausgewählt haben, werden die Traces nach der Antwortzeit des Services sortiert.

7. Wählen Sie im Navigationsbereich der CloudWatch Konsole SloS aus. Sie sehen die SLOs, die das Skript für die Beispielanwendung erstellt hat. Weitere Informationen zu SLOs finden Sie unter Servicelevel-Ziele (SLOs).

(Optional) Schritt 5: Bereinigen

Wenn Sie mit dem Testen von Application Signals fertig sind, können Sie ein von Amazon bereitgestelltes Skript verwenden, um die in Ihrem Konto für die Beispielanwendung erstellten Artefakte zu bereinigen und zu löschen. Geben Sie den folgenden Befehl ein, um die Bereinigung durchzuführen. *new-cluster-name*Ersetzen Sie durch den Namen des Clusters, den Sie für die Beispiel-App erstellt haben, und ersetzen Sie *region* -name durch den Namen der AWS Region, z. B. `us-west-1`

```
cd application-signals-demo/scripts/eks/appsignals/one-step && ./cleanup.sh new-
cluster-name region-name
```

Application Signals auf anderen Plattformen mit einer benutzerdefinierten Konfiguration aktivieren

⚠ Application Signals befindet sich in der Vorschauversion. Wenn Sie Feedback zu dieser Funktion haben, können Sie uns unter app-signals-feedback@amazon .com kontaktieren.

Aktivieren Sie CloudWatch Application Signals auf anderen Plattformen als Amazon EKS, indem Sie die benutzerdefinierten Einrichtungsschritte in diesen Abschnitten verwenden. Auf diesen Architekturen installieren und konfigurieren Sie den CloudWatch Agenten und die AWS Distribution für OpenTelemetry sich selbst.

Auf diesen Architekturen erkennt Application Signals die Namen Ihrer Services oder deren Cluster oder Hosts nicht automatisch. Sie müssen diese Namen bei der benutzerdefinierten Einrichtung angeben, und die Namen, die Sie angeben, werden auf den Dashboards von Application Signals angezeigt.

Themen

- Eine benutzerdefinierte Einrichtung verwenden, um Application Signals auf Amazon ECS zu aktivieren

- Eine benutzerdefinierte Einrichtung verwenden, um Application Signals auf Amazon EC2 und anderen Plattformen zu aktivieren

Eine benutzerdefinierte Einrichtung verwenden, um Application Signals auf Amazon ECS zu aktivieren

⚠ Application Signals befindet sich in der Vorschauversion. Wenn Sie Feedback zu dieser Funktion haben, können Sie uns unter app-signals-feedback@amazon .com kontaktieren.

Verwenden Sie diese Anweisungen zur benutzerdefinierten Einrichtung, um Ihre Anwendungen auf Amazon ECS in CloudWatch Application Signals zu integrieren. Sie installieren und konfigurieren den CloudWatch Agenten und die AWS Distribution für OpenTelemetry sich selbst.

Auf Amazon-ECS-Clustern erkennt Application Signals die Namen Ihrer Services oder der Cluster, in denen sie ausgeführt werden, nicht automatisch. Sie müssen diese Namen bei der benutzerdefinierten Einrichtung angeben, und die Namen, die Sie angeben, werden auf den Dashboards von Application Signals angezeigt.

⚠ Important

Nur der Netzwerkmodus `awsvpc` wird unterstützt.

Schritt 1: Application Signals in Ihrem Konto aktivieren

Wenn Sie Application Signals in diesem Konto noch nicht aktiviert haben, müssen Sie Application Signals die Berechtigungen gewähren, die es benötigt, um Ihre Services zu erkennen. Gehen Sie dazu wie folgt vor. Dies muss nur einmal für Ihr Konto durchgeführt werden.

So aktivieren Sie Application Signals für Ihre Anwendungen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Services.
3. Wählen Sie Mit der Entdeckung Ihrer Services beginnen.
4. Aktivieren Sie das Kontrollkästchen und wählen Sie Mit der Entdeckung von Services beginnen.

Wenn Sie diesen Schritt zum ersten Mal in Ihrem Konto ausführen, wird die AWSServiceRoleForCloudWatchApplicationSignalsdienstverknüpfte Rolle erstellt. Diese Rolle gewährt Application Signals die folgenden Berechtigungen:

- `xray:GetServiceGraph`
- `logs:StartQuery`
- `logs:GetQueryResults`
- `cloudwatch:GetMetricData`
- `cloudwatch:ListMetrics`
- `tag:GetResources`

Weitere Informationen über diese Rolle finden Sie unter Dienstbezogene Rollenberechtigungen für Anwendungssignale CloudWatch .

Schritt 2: IAM-Rollen erstellen

Sie müssen zwei IAM-Rollen erstellen. Wenn Sie diese Rollen bereits erstellt haben, müssen Sie ihnen möglicherweise Berechtigungen hinzufügen.

- ECS-Aufgabenrolle – Container verwenden diese Rolle zum Ausführen. Die Berechtigungen sollten den Anforderungen Ihrer Anwendungen entsprechen, plus CloudWatchAgentServerPolicyund AWSXRayWriteOnlyAccess.
- ECS-Aufgabenausführungsrolle – Amazon ECS verwendet diese Rolle, um Ihre Container zu starten und auszuführen. Wenn Sie diese Rolle bereits erstellt haben, fügen Sie ihr die AmazonSSM -ReadOnlyAccess, AmazonECS TaskExecutionRolePolicy - und CloudWatchAgentServerPolicyRichtlinien hinzu.

  Wenn Sie vertraulichere Daten für Amazon ECS speichern müssen, finden Sie weitere Informationen unter Angabe sensibler Daten.

Weitere Informationen zum Erstellen von IAM-Rollen finden Sie unter Erstellen von IAM-Rollen.

Schritt 3: Bereiten Sie die Agentenkonfiguration vor CloudWatch

Bereiten Sie zunächst die Agentenkonfiguration mit Application Signals aktiviert vor. Erstellen Sie dazu eine lokale Datei mit dem Namen `/tmp/ecs-cwagent.json`.

```
{
  "traces": {
    "traces_collected": {
      "app_signals": {}
    }
  },
  "logs": {
    "metrics_collected": {
      "app_signals": {}
    }
  }
}
```

Laden Sie dann diese Konfiguration in den SSM-Parameterspeicher hoch. Geben Sie dazu den folgenden Befehl ein. Ersetzen Sie in der Datei *$REGION* durch Ihren tatsächlichen Regionsnamen.

```
aws ssm put-parameter \
--name "ecs-cwagent" \
--type "String" \
--value "`cat /tmp/ecs-cwagent.json`" \
--region "$REGION"
```

Schritt 4: Instrumentieren Sie Ihre Anwendung mit dem CloudWatch Agenten

Der nächste Schritt besteht darin, Ihre Anwendung für CloudWatch Application Signals zu instrumentieren.

Java

Um Ihre Anwendung auf Amazon ECS mit dem CloudWatch Agenten zu instrumentieren

1. Geben Sie zunächst einen Bind-Mount an. Das Volume wird in den nächsten Schritten verwendet, um Dateien containerübergreifend freizugeben. Sie werden diesen Bind-Mount später in diesem Verfahren verwenden.

   ```
   "volumes": [
     {
       "name": "opentelemetry-auto-instrumentation"
     }
   ]
   ```

2. Fügen Sie eine CloudWatch Sidecar-Definition für Agenten hinzu. Fügen Sie dazu einen neuen Container namens `ecs-cwagent` an die Aufgabendefinition Ihrer Anwendung an. Ersetzen Sie *$REGION* durch Ihren tatsächlichen Regionsnamen. Ersetzen Sie durch den Pfad zum neuesten CloudWatch Container-Image in Amazon Elastic Container Registry. Weitere Informationen finden Sie unter cloudwatch-agent auf Amazon ECR.

```
{
  "name": "ecs-cwagent",
  "image": "$IMAGE",
  "essential": true,
  "secrets": [
    {
      "name": "CW_CONFIG_CONTENT",
      "valueFrom": "ecs-cwagent"
    }
  ],
  "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
      "awslogs-create-group": "true",
      "awslogs-group": "/ecs/ecs-cwagent",
      "awslogs-region": "$REGION",
      "awslogs-stream-prefix": "ecs"
    }
  }
}
```

3. Fügen Sie einen neuen Container namens `init` an die Aufgabendefinition Ihrer Anwendung an. Ersetzen Sie *$IMAGE* durch das neueste Bild aus dem AWS Distro for OpenTelemetry Amazon ECR-Image-Repository.

```
{
  "name": "init",
  "image": "$IMAGE",
  "essential": false,
  "command": [
    "cp",
    "/javaagent.jar",
    "/otel-auto-instrumentation/javaagent.jar"
  ],
  "mountPoints": [
    {
```

```
      "sourceVolume": "opentelemetry-auto-instrumentation",
      "containerPath": "/otel-auto-instrumentation",
      "readOnly": false
    }
  ]
}
```

4. Fügen Sie die folgenden Umgebungsvariablen Ihrem Anwendungs-Container hinzu. Weitere Informationen finden Sie unter

Umgebungsvariable	Einstellen zur Aktivierung von Application Signals
`OTEL_RESOURCE_ATTRIBUTES`	Ersetzen Sie `$SVC_NAME` mit dem Namen Ihrer Anwendung. Dies wird in den Dashboards von Application Signals als Name der Anwendung angezeigt. Ersetzen Sie `$HOST_ENV` durch die Host-Umgebung, in der Ihre Anwendung ausgeführt wird. Dies wird in den Dashboards von Application Signals als Gehostet in-Umgebung Ihrer Anwendung angezeigt.
`OTEL_AWS_APP_SIGNALS_ENABLED`	Stellen Sie auf ein, `true` um die Anwendungssignale zu aktivieren. SpanMetricsProcessor
`OTEL_METRICS_EXPORTER`	Stellen Sie auf `none` ein, um andere Metrik-Exportprogramme zu deaktivieren.
`OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`	Auf einstellen, `http://127.0.0.1:4315` um Messwerte an den CloudWatch Beiwagen zu senden.
`OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`	Auf einstellen, `http://127.0.0.1:4315` um Traces an den Beiwagen zu senden. CloudWatch

Umgebungsvariable	Einstellen zur Aktivierung von Application Signals
OTEL_TRACES_SAMPLER	Definieren Sie X-Ray als den Traces-Sammler.
OTEL_PROPAGATORS	Fügen Sie X-Ray als einen der Propagatoren hinzu.
JAVA_TOOL_OPTIONS	Injizieren Sie den Agenten AWS Distro for Java. OpenTelemetry

5. Mounten Sie das Volume `opentelemetry-auto-instrumentation`, das Sie in Schritt 1 dieses Verfahrens definiert haben.

   Verwenden Sie für eine Java-Anwendung Folgendes.

```
{
  "name": "app",
   ...
  "environment": [
    {
      "name": "OTEL_RESOURCE_ATTRIBUTES",
      "value": "aws.hostedin.environment=$HOST_ENV,service.name=$SVC_NAME"
    },
    {
      "name": "OTEL_AWS_APP_SIGNALS_ENABLED",
      "value": "true"
    },
    {
      "name": "OTEL_METRICS_EXPORTER",
      "value": "none"
    },
    {
      "name": "JAVA_TOOL_OPTIONS",
      "value": " -javaagent:/otel-auto-instrumentation/javaagent.jar"
    },
    {
      "name": "OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT",
      "value": "http://127.0.0.1:4315"
    },
    {
```

```
      "name": "OTEL_TRACES_SAMPLER",
      "value": "xray"
    },
    {
      "name": "OTEL_EXPORTER_OTLP_TRACES_ENDPOINT",
      "value": "http://127.0.0.1:4315"
    },
    {
      "name": "OTEL_PROPAGATORS",
      "value": "tracecontext,baggage,b3,xray"
    }
  ],
  "mountPoints": [
    {
      "sourceVolume": "opentelemetry-auto-instrumentation",
      "containerPath": "/otel-auto-instrumentation",
      "readOnly": false
    }
  ]
}
```

Python

Bevor Sie Application Signals für Ihre Python-Anwendungen aktivieren, sollten Sie die folgenden Überlegungen beachten.

- In einigen containerisierten Anwendungen kann eine fehlende `PYTHONPATH` Umgebungsvariable manchmal dazu führen, dass die Anwendung nicht gestartet werden kann. Um dieses Problem zu beheben, stellen Sie sicher, dass Sie die `PYTHONPATH` Umgebungsvariable auf den Speicherort des Arbeitsverzeichnisses Ihrer Anwendung setzen. Dies ist auf ein bekanntes Problem mit der OpenTelemetry automatischen Instrumentierung zurückzuführen. Weitere Informationen zu diesem Problem finden Sie unter Python-Autoinstrumentation-Einstellung von PYTHONPATH ist nicht kompatibel.
- Für Django-Anwendungen sind zusätzliche Konfigurationen erforderlich, die in der OpenTelemetry Python-Dokumentation beschrieben werden.
  - Verwenden Sie das `--noreload` Flag, um ein automatisches Neuladen zu verhindern.
  - Setzen Sie die `DJANGO_SETTINGS_MODULE` Umgebungsvariable auf den Speicherort der Datei Ihrer Django-Anwendung. `settings.py` Dadurch wird sichergestellt, dass

OpenTelemetry Sie korrekt auf Ihre Django-Einstellungen zugreifen und diese integrieren können.

Um Ihre Python-Anwendung auf Amazon ECS mit dem CloudWatch Agenten zu instrumentieren

1. Geben Sie zunächst einen Bind-Mount an. Das Volume wird in den nächsten Schritten verwendet, um Dateien containerübergreifend freizugeben. Sie werden diesen Bind-Mount später in diesem Verfahren verwenden.

```
"volumes": [
  {
    "name": "opentelemetry-auto-instrumentation-python"
  }
]
```

2. Fügen Sie eine CloudWatch Agent-Sidecar-Definition hinzu. Fügen Sie dazu einen neuen Container namens `ecs-cwagent` an die Aufgabendefinition Ihrer Anwendung an. Ersetzen Sie *$REGION* durch Ihren tatsächlichen Regionsnamen. Ersetzen Sie durch den Pfad zum neuesten CloudWatch Container-Image in Amazon Elastic Container Registry. Weitere Informationen finden Sie unter cloudwatch-agent auf Amazon ECR.

```
{
  "name": "ecs-cwagent",
  "image": "$IMAGE",
  "essential": true,
  "secrets": [
    {
      "name": "CW_CONFIG_CONTENT",
      "valueFrom": "ecs-cwagent"
    }
  ],
  "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
      "awslogs-create-group": "true",
      "awslogs-group": "/ecs/ecs-cwagent",
      "awslogs-region": "$REGION",
      "awslogs-stream-prefix": "ecs"
    }
  }
```

```
}
```

3. Fügen Sie einen neuen Container namens `init` an die Aufgabendefinition Ihrer Anwendung an. Ersetzen Sie *`$IMAGE`* durch das neueste Bild aus dem AWS Distro for OpenTelemetry Amazon ECR-Image-Repository.

```
{
    "name": "init",
    "image": "$IMAGE",
    "essential": false,
    "command": [
        "cp",
        "-a",
        "/autoinstrumentation/.",
        "/otel-auto-instrumentation-python"
    ],
    "mountPoints": [
        {
            "sourceVolume": "opentelemetry-auto-instrumentation-python",
            "containerPath": "/otel-auto-instrumentation-python",
            "readOnly": false
        }
    ]
}
```

4. Fügen Sie die folgenden Umgebungsvariablen Ihrem Anwendungs-Container hinzu. Weitere Informationen finden Sie unter

Umgebungsvariable	Einstellen zur Aktivierung von Application Signals
`OTEL_RESOURCE_ATTRIBUTES`	Ersetzen Sie `$SVC_NAME` mit dem Namen Ihrer Anwendung. Dies wird in den Dashboards von Application Signals als Name der Anwendung angezeigt. Ersetzen Sie `$HOST_ENV` durch die Host-Umgebung, in der Ihre Anwendung ausgeführt wird. Dies wird in den Dashboards von Application Signals als

Umgebungsvariable	Einstellen zur Aktivierung von Application Signals
	Gehostet in-Umgebung Ihrer Anwendung angezeigt.
`OTEL_AWS_APP_SIGNALS_ENABLED`	Stellen Sie auf ein, `true` um die Anwendungssignale zu aktivieren. SpanMetricsProcessor
`OTEL_METRICS_EXPORTER`	Stellen Sie auf `none` ein, um andere Metrik-Exportprogramme zu deaktivieren.
`OTEL_EXPORTER_OTLP_PROTOCOL`	Legt fest, `http/protobuf` dass Metriken und Traces CloudWatch über HTTP gesendet werden sollen.
`OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`	Auf einstellen, `http://127.0.0.1:4316/v1/metrics` um Metriken an den CloudWatch Beiwagen zu senden.
`OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`	Auf einstellen, `http://127.0.0.1:4316/v1/traces` um Traces an den Beiwagen zu senden. CloudWatch
`OTEL_TRACES_SAMPLER`	Definieren Sie X-Ray als den Traces-Sammler.
`OTEL_PROPAGATORS`	Fügen Sie X-Ray als einen der Propagatoren hinzu.
`OTEL_PYTHON_DISTRO`	Auf einstellen, `aws_distro` um die ADOT-Python-Instrumentierung zu verwenden.
`OTEL_PYTHON_CONFIGURATOR`	Auf einstellen, `aws_configuration` um die ADOT-Python-Konfiguration zu verwenden.

Umgebungsvariable	Einstellen zur Aktivierung von Application Signals
PYTHONPATH	$APP_PATH Ersetzen Sie durch den Speicherort des Arbeitsverzeichnisses der Anwendung innerhalb des Containers. Dies ist erforderlich, damit der Python-Interpreter Ihre Anwendungsmodule finden kann.
DJANGO_SETTINGS_MODULE	Nur für Django-Anwendungen erforderlich. Stellen Sie es auf den Speicherort der Datei Ihrer Django-Anwendung einsettings.py . Ersetzen$PATH_TO_SETTINGS .

5. Mounten Sie das Volume opentelemetry-auto-instrumentation-python, das Sie in Schritt 1 dieses Verfahrens definiert haben.

   Verwenden Sie für eine Python-Anwendung Folgendes.

```
{
    "name": "app",
    ...
    "environment": [
        {
            "name": "PYTHONPATH",
            "value": "/otel-auto-instrumentation-python/opentelemetry/
instrumentation/auto_instrumentation:$APP_PATH:/otel-auto-instrumentation-
python"
        },
        {
            "name": "OTEL_EXPORTER_OTLP_PROTOCOL",
            "value": "http/protobuf"
        },
        {
            "name": "OTEL_TRACES_SAMPLER",
            "value": "xray"
        },
        {
            "name": "OTEL_TRACES_SAMPLER_ARG",
            "value": "endpoint=http://localhost:2000"
        },
```

```
        {
            "name": "OTEL_LOGS_EXPORTER",
            "value": "none"
        },
        {
            "name": "OTEL_PYTHON_DISTRO",
            "value": "aws_distro"
        },
        {
            "name": "OTEL_PYTHON_CONFIGURATOR",
            "value": "aws_configurator"
        },
        {
            "name": "OTEL_EXPORTER_OTLP_TRACES_ENDPOINT",
            "value": "http://localhost:4316/v1/traces"
        },
        {
            "name": "OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT",
            "value": "http://localhost:4316/v1/metrics"
        },
        {
            "name": "OTEL_METRICS_EXPORTER",
            "value": "none"
        },
        {
            "name": "OTEL_AWS_APP_SIGNALS_ENABLED",
            "value": "true"
        },
        {
            "name": "OTEL_RESOURCE_ATTRIBUTES",
            "value": "aws.hostedin.environment=$HOST_ENV,service.name=$SVC_NAM"
        },
        {
            "name": "DJANGO_SETTINGS_MODULE",
            "value": "$PATH_TO_SETTINGS.settings"
        }
    ],
    "mountPoints": [
        {
            "sourceVolume": "opentelemetry-auto-instrumentation-python",
            "containerPath": "/otel-auto-instrumentation-python",
            "readOnly": false
        }
    ]
```

```
}
```

Schritt 5: Ihre Anwendung bereitstellen

Erstellen Sie eine neue Version Ihrer Aufgabendefinition und stellen Sie sie in Ihrem Anwendungscluster bereit. In der neu erstellten Aufgabe sollten Sie drei Container sehen:

- `init`
- `ecs-cwagent`
- `app`

Eine benutzerdefinierte Einrichtung verwenden, um Application Signals auf Amazon EC2 und anderen Plattformen zu aktivieren

⚠ Application Signals befindet sich in der Vorschauversion. Wenn Sie Feedback zu dieser Funktion haben, können Sie uns unter app-signals-feedback@amazon .com kontaktieren.

Für Anwendungen, die auf Amazon EC2 und anderen Architekturen ausgeführt werden, die nicht Amazon EKS sind, installieren und konfigurieren Sie den CloudWatch Agenten und die AWS Distribution selbst. OpenTelemetry Auf diesen Architekturen, die mit einer benutzerdefinierten Einrichtung von Application Signals aktiviert wurden, erkennt Application Signals die Namen Ihrer Services oder deren Cluster oder Hosts nicht automatisch. Sie müssen diese Namen bei der benutzerdefinierten Einrichtung angeben, und die Namen, die Sie angeben, werden auf den Dashboards von Application Signals angezeigt.

Die folgenden Schritte wurden auf Amazon EC2 EC2-Instances getestet, es wird jedoch erwartet, dass sie auch auf anderen Architekturen funktionieren, für die AWS Distro unterstützt wird. OpenTelemetry

Voraussetzungen

- Um Support für Application Signals zu erhalten, müssen Sie die neueste Version sowohl des Agenten als auch der CloudWatch Distribution for Agent verwenden. AWS OpenTelemetry

- Sie müssen den auf der Instanz AWS CLI installiert haben. Wir empfehlen AWS CLI Version 2, aber Version 1 sollte auch funktionieren. Weitere Informationen zur Installation von finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI. AWS CLI

> **Important**
>
> Wenn Sie bereits eine Anwendung verwenden OpenTelemetry , die Sie für Application Signals aktivieren möchten, finden Sie weitere Informationen, OpenTelemetry Überlegungen zur Kompatibilität bevor Sie Application Signals aktivieren.

Schritt 1: Application Signals in Ihrem Konto aktivieren

Wenn Sie Application Signals in diesem Konto noch nicht aktiviert haben, müssen Sie Application Signals die Berechtigungen gewähren, die es benötigt, um Ihre Services zu erkennen. Gehen Sie dazu wie folgt vor. Dies muss nur einmal für Ihr Konto durchgeführt werden.

So aktivieren Sie Application Signals für Ihre Anwendungen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Services.
3. Wählen Sie Mit der Entdeckung Ihrer Services beginnen.
4. Aktivieren Sie das Kontrollkästchen und wählen Sie Mit der Entdeckung von Services beginnen.

   Wenn Sie diesen Schritt zum ersten Mal in Ihrem Konto ausführen, wird die AWSServiceRoleForCloudWatchApplicationSignalsdienstverknüpfte Rolle erstellt. Diese Rolle gewährt Application Signals die folgenden Berechtigungen:

   - `xray:GetServiceGraph`
   - `logs:StartQuery`
   - `logs:GetQueryResults`
   - `cloudwatch:GetMetricData`
   - `cloudwatch:ListMetrics`
   - `tag:GetResources`

Weitere Informationen über diese Rolle finden Sie unter Dienstbezogene Rollenberechtigungen für Anwendungssignale CloudWatch .

Schritt 2: Laden Sie den Agenten herunter und starten Sie ihn CloudWatch

Um den CloudWatch Agenten im Rahmen der Aktivierung von Application Signals auf einer Amazon EC2 EC2-Instance zu installieren

1. Laden Sie die neueste Version des CloudWatch Agenten auf die Instance herunter. Wenn der CloudWatch Agent auf der Instanz bereits installiert ist, müssen Sie ihn möglicherweise aktualisieren. Nur Versionen des Agenten, die am 30. November 2023 oder später veröffentlicht wurden, unterstützen CloudWatch Application Signals.

   Informationen zum Herunterladen des CloudWatch Agenten finden Sie unterLaden Sie das CloudWatch Agentenpaket herunter.
2. Bevor Sie den CloudWatch Agenten starten, konfigurieren Sie ihn so, dass er Application Signals aktiviert. Das folgende Beispiel zeigt eine CloudWatch Agentenkonfiguration, die Application Signals sowohl für Metriken als auch für Traces auf einem EC2-Host aktiviert.

   Sie können diese Datei mit dem folgenden Befehl erstellen:

   ```
   vim amazon-cloudwatch-agent.json
   ```

   Fügen Sie der Datei die folgenden Inhalte hinzu.

   ```
   {
     "traces": {
       "traces_collected": {
         "app_signals": {}
       }
     },
     "logs": {
       "metrics_collected": {
         "app_signals": {}
       }
     }
   }
   ```

3. Fügen Sie die CloudWatchAgentServerPolicyund AWSXrayWriteOnlyAccessIAM-Richtlinien der IAM-Rolle Ihrer Amazon EC2 EC2-Instance hinzu.

   a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.

   b. Wählen Sie Rollen und suchen Sie nach der Rolle, die von Ihrer Amazon-EC2-Instance verwendet wird. Wählen Sie dann den Namen dieser Rolle.

   c. Wählen Sie auf der Registerkarte Berechtigungen die Option Berechtigungen hinzufügen und dann Richtlinien anfügen.

   d. Finden Sie CloudWatchAgentServerPolicy. Verwenden Sie bei Bedarf das Suchfeld. Aktivieren Sie dann das Kontrollkästchen für die Richtlinie und wählen Sie dann Berechtigungen hinzufügen.

   e. Finden AWSXrayWriteOnlyAccess. Verwenden Sie bei Bedarf das Suchfeld. Aktivieren Sie dann das Kontrollkästchen für die Richtlinie und wählen Sie dann Berechtigungen hinzufügen.

4. Starten Sie den CloudWatch Agenten, indem Sie die folgenden Befehle eingeben. *agent-config-file-path*Ersetzen Sie durch den Pfad zur CloudWatch Agentenkonfigurationsdatei, z. `./amazon-cloudwatch-agent.json` B. Sie müssen das `file:`-Präfix wie abgebildet angeben.

```
export CONFIG_FILE_PATH=./amazon-cloudwatch-agent.json
```

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl \
-a fetch-config \
-m ec2 -s -c file:$CONFIG_FILE_PATH
```

Schritt 3: Ihre Anwendung instrumentieren und starten

Der nächste Schritt besteht darin, Ihre Anwendung für CloudWatch Application Signals zu instrumentieren.

Java

Um Ihre Java-Anwendungen im Rahmen der Aktivierung von Application Signals auf einer Amazon EC2 EC2-Instance zu instrumentieren

1. Laden Sie die neueste Version des Autoinstrumentation-Agenten von AWS Distro for OpenTelemetry Java herunter. Sie können die neueste Version über diesen Link herunterladen. Informationen zu allen veröffentlichten Versionen finden Sie unter aws-otel-java-instrumentation Releases.
2. Um die Vorteile von Application Signals zu optimieren, verwenden Sie Umgebungsvariablen, um zusätzliche Informationen bereitzustellen, bevor Sie Ihre Anwendung starten. Diese Informationen werden in den Dashboards von Application Signals angezeigt.

   a. Geben Sie für die `OTEL_RESOURCE_ATTRIBUTES`-Variable die folgenden Informationen als Schlüssel-Wert-Paare an:

      - `aws.hostedin.environment` legt die Umgebung fest, in der die Anwendung ausgeführt wird. Dies wird in den Dashboards von Application Signals als Gehostet in-Umgebung Ihrer Anwendung angezeigt. Dieser Attributschlüssel wird nur von Application Signals verwendet und in Röntgen-Trace-Anmerkungen und CloudWatch metrische Dimensionen umgewandelt. Wenn Sie keinen Wert für diesen Schlüssel angeben, wird der Standardwert von `Generic` verwendet.
      - `service.name` legt den Namen des Services fest. Dies wird in den Dashboards von Application Signals als Servicename der Anwendung angezeigt. Wenn Sie keinen Wert für diesen Schlüssel angeben, wird der Standardwert von `unknown_service` verwendet.

   b. Geben Sie für die `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`-Variable die Basis-Endpunkt-URL an, in die die Traces exportiert werden sollen. Der CloudWatch Agent macht 4315 als seinen OLTP-Port verfügbar. Da Anwendungen in Amazon EC2 mit dem lokalen CloudWatch Agenten kommunizieren, sollten Sie diesen Wert auf festlegen `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4315`

   c. Geben Sie für die `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`-Variable die Basis-Endpunkt-URL an, in die die Metriken exportiert werden sollen. Der CloudWatch Agent gibt 4315 als seinen OLTP-Port an. Da Anwendungen in Amazon EC2 mit dem lokalen CloudWatch Agenten kommunizieren, sollten Sie diesen Wert auf festlegen `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4315`

d. Geben Sie für die `JAVA_TOOL_OPTIONS` Variable den Pfad an, in dem der Autoinstrumentation-Agent von AWS Distro for OpenTelemetry Java gespeichert ist.

```
export JAVA_TOOL_OPTIONS=' -javaagent:$ADOT_AGENT_PATH'
```

Beispielsweise:

```
export ADOT_AGENT_PATH=./aws-opentelemetry-agent.jar
```

e. Für die `OTEL_METRICS_EXPORTER`-Variable empfehlen wir, den Wert auf `none` einzustellen. Dadurch werden andere Metrik-Exportprogramme deaktiviert, sodass nur der Application-Signals-Exporter verwendet wird.

f. Aktivieren Sie SpanMetricProcessor (SMP) für die `OTEL_AWS_APP_SIGNALS_ENABLED` Variable, indem Sie auf setzen. `OTEL_AWS_APP_SIGNALS_ENABLED` `true` Dadurch werden Application-Signals-Metriken aus Traces generiert.

3. Starten Sie Ihre Anwendung mit den Umgebungsvariablen, die im vorherigen Schritt beschrieben wurden. Im Folgenden finden Sie ein Beispiel für ein Start-Skript.

```
JAVA_TOOL_OPTIONS=' -javaagent:$ADOT_AGENT_PATH' \
OTEL_METRICS_EXPORTER=none \
OTEL_AWS_APP_SIGNALS_ENABLED=true \
OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4315 \
OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4315 \
OTEL_RESOURCE_ATTRIBUTES=aws.hostedin.environment=$YOUR_HOST_ENV,service.name=
$YOUR_SVC_NAME \
java -jar $MY_JAVA_APP.jar
```

Python

Um Ihre Python-Anwendungen im Rahmen der Aktivierung von Application Signals auf einer Amazon EC2 EC2-Instance zu instrumentieren

1. Laden Sie die neueste Version des Autoinstrumentation-Agenten von AWS Distro for OpenTelemetry Python herunter. Für die Installation führen Sie den folgenden -Befehl aus.

```
pip install aws-opentelemetry-distro
```

Sie können Informationen zu allen veröffentlichten Versionen unter AWS Distro for OpenTelemetry Python Instrumentation einsehen.

2. Um die Vorteile von Application Signals zu optimieren, verwenden Sie Umgebungsvariablen, um zusätzliche Informationen bereitzustellen, bevor Sie Ihre Anwendung starten. Diese Informationen werden in den Dashboards von Application Signals angezeigt.

   a. Geben Sie für die `OTEL_RESOURCE_ATTRIBUTES`-Variable die folgenden Informationen als Schlüssel-Wert-Paare an:

      - `aws.hostedin.environment` legt die Umgebung fest, in der die Anwendung ausgeführt wird. Dies wird in den Dashboards von Application Signals als Gehostet in-Umgebung Ihrer Anwendung angezeigt. Dieser Attributschlüssel wird nur von Application Signals verwendet und in Röntgen-Trace-Anmerkungen und CloudWatch metrische Dimensionen umgewandelt. Wenn Sie keinen Wert für diesen Schlüssel angeben, wird der Standardwert von `Generic` verwendet.
      - `service.name` legt den Namen des Services fest. Dies wird in den Dashboards von Application Signals als Servicename der Anwendung angezeigt. Wenn Sie keinen Wert für diesen Schlüssel angeben, wird der Standardwert von `unknown_service` verwendet.

   b. Geben Sie für die `OTEL_EXPORTER_OTLP_PROTOCOL` Variable `http/protobuf` an, dass Telemetriedaten über HTTP an die in den folgenden Schritten aufgeführten CloudWatch Agenten-Endpunkte exportiert werden sollen.

   c. Geben Sie für die `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`-Variable die Basis-Endpunkt-URL an, in die die Traces exportiert werden sollen. Der CloudWatch Agent stellt 4316 als seinen OLTP-Port über HTTP zur Verfügung. Da Anwendungen in Amazon EC2 mit dem lokalen CloudWatch Agenten kommunizieren, sollten Sie diesen Wert auf festlegen `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4316/v1/traces`

   d. Geben Sie für die `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`-Variable die Basis-Endpunkt-URL an, in die die Metriken exportiert werden sollen. Der CloudWatch Agent stellt 4316 als seinen OLTP-Port über HTTP zur Verfügung. Da Anwendungen in Amazon EC2 mit dem lokalen CloudWatch Agenten kommunizieren, sollten Sie diesen Wert auf festlegen `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4316/v1/metrics`

e. Für die OTEL_METRICS_EXPORTER-Variable empfehlen wir, den Wert auf none einzustellen. Dadurch werden andere Metrik-Exportprogramme deaktiviert, sodass nur der Application-Signals-Exporter verwendet wird.

f. Aktivieren Sie für die OTEL_AWS_APP_SIGNALS_ENABLED Variable die Einstellung SpanMetricProcessor by OTEL_AWS_APP_SIGNALS_ENABLED auftrue. Dadurch werden Application-Signals-Metriken aus Traces generiert.

3. Starten Sie Ihre Anwendung mit den Umgebungsvariablen, die im vorherigen Schritt beschrieben wurden. Im Folgenden finden Sie ein Beispiel für ein Start-Skript.

   - Ersetzen Sie $HOST_ENV durch die Host-Umgebung, in der Ihre Anwendung ausgeführt wird. Dies wird in den Dashboards von Application Signals als Hosted In-Umgebung für Ihre Anwendung angezeigt.
   - Ersetzen Sie $SVC_NAME mit dem Namen Ihrer Anwendung. Dies wird in den Dashboards von Application Signals als Name der Anwendung angezeigt.
   - $PYTHON_APPErsetzen Sie es durch den Speicherort und den Namen Ihrer Anwendung.

```
OTEL_METRICS_EXPORTER=none \
OTEL_LOGS_EXPORTER=none \
OTEL_AWS_APP_SIGNALS_ENABLED=true \
OTEL_PYTHON_DISTRO=aws_distro \
OTEL_PYTHON_CONFIGURATOR=aws_configurator \
OTEL_EXPORTER_OTLP_PROTOCOL=http/protobuf \
OTEL_TRACES_SAMPLER=xray \
OTEL_TRACES_SAMPLER_ARG="endpoint=http://localhost:2000" \
OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4316/v1/metrics \
OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4316/v1/traces \
OTEL_RESOURCE_ATTRIBUTES=aws.hostedin.environment=$HOST_ENV,service.name=
$SVC_NAME \
opentelemetry-instrument python $PYTHON_APP.py
```

Bevor Sie Application Signals für Ihre Python-Anwendungen aktivieren, sollten Sie die folgenden Überlegungen beachten.

- In einigen containerisierten Anwendungen kann eine fehlende PYTHONPATH Umgebungsvariable manchmal dazu führen, dass die Anwendung nicht gestartet werden kann. Um dieses Problem zu beheben, stellen Sie sicher, dass Sie die PYTHONPATH Umgebungsvariable auf den Speicherort des Arbeitsverzeichnisses Ihrer Anwendung

setzen. Dies ist auf ein bekanntes Problem mit der OpenTelemetry automatischen Instrumentierung zurückzuführen. Weitere Informationen zu diesem Problem finden Sie unter Python-Autoinstrumentation-Einstellung von PYTHONPATH ist nicht kompatibel.

- Für Django-Anwendungen sind zusätzliche Konfigurationen erforderlich, die in der OpenTelemetry Python-Dokumentation beschrieben werden.
  - Verwenden Sie das `--noreload` Flag, um ein automatisches Neuladen zu verhindern.
  - Setzen Sie die `DJANGO_SETTINGS_MODULE` Umgebungsvariable auf den Speicherort der Datei Ihrer Django-Anwendung. `settings.py` Dadurch wird sichergestellt, dass OpenTelemetry Sie korrekt auf Ihre Django-Einstellungen zugreifen und diese integrieren können.

Fehlerbehebung bei der Installation von Application Signals

⚠ Application Signals befindet sich in der Vorschauversion. Wenn Sie Feedback zu dieser Funktion haben, können Sie uns unter app-signals-feedback@amazon .com kontaktieren.

Dieser Abschnitt enthält Tipps zur Fehlerbehebung bei CloudWatch Anwendungssignalen.

Themen

Die Anwendung wird nicht gestartet, nachdem Application Signals aktiviert wurde

Wenn Ihre Anwendung auf einem Amazon-EKS-Cluster nicht startet, nachdem Sie Application Signals auf dem Cluster aktiviert haben, überprüfen Sie Folgendes:

- Prüfen Sie, ob die Anwendung durch eine andere Überwachungslösung instrumentiert wurde. Application Signals unterstützt nicht die Koexistenz mit anderen Instrumentierungslösungen.

- Vergewissern Sie sich, dass Ihre Anwendung die Kompatibilitätsanforderungen für die Verwendung von Application Signals erfüllt. Weitere Informationen finden Sie unter Application Signals, unterstützte Systeme .
- Wenn Ihre Anwendung die Application Signal-Artefakte wie den Agenten und CloudWatch die Agent-Images von AWS Distro for OpenTelemetery Java oder Python nicht abrufen konnte, liegt möglicherweise ein Netzwerkproblem vor.

Um das Problem zu beheben, entfernen Sie die Anmerkung `instrumentation.opentelemetry.io/inject-java: "true"` oder `instrumentation.opentelemetry.io/inject-python: "true"` aus Ihrem Anwendungsbereitstellungsmanifest und stellen Sie Ihre Anwendung erneut bereit. Überprüfen Sie dann, ob die Anwendung funktioniert.

Die Python-Anwendung startet nicht, nachdem Application Signals aktiviert wurde

Es ist ein bekanntes Problem bei der OpenTelemetry automatischen Instrumentierung, dass eine fehlende `PYTHONPATH` Umgebungsvariable manchmal dazu führen kann, dass die Anwendung nicht gestartet werden kann. Um dieses Problem zu lösen, stellen Sie sicher, dass Sie die `PYTHONPATH` Umgebungsvariable auf den Speicherort des Arbeitsverzeichnisses Ihrer Anwendung setzen. Weitere Informationen zu diesem Problem finden Sie unter Die Python-Autoinstrumentation-Einstellung von PYTHONPATH entspricht nicht dem Modulauflösungsverhalten von Python, wodurch Django-Anwendungen beschädigt werden.

Für Django-Anwendungen sind zusätzliche Konfigurationen erforderlich, die in der OpenTelemetry Python-Dokumentation beschrieben werden.

- Verwenden Sie das `--noreload` Flag, um ein automatisches Neuladen zu verhindern.
- Setzen Sie die `DJANGO_SETTINGS_MODULE` Umgebungsvariable auf den Speicherort der Datei Ihrer Django-Anwendung. `settings.py` Dadurch wird sichergestellt, dass OpenTelemetry Sie korrekt auf Ihre Django-Einstellungen zugreifen und diese integrieren können.

Telemetriedaten fehlen in CloudWatch und X-Ray

Wenn in den Dashboards von Application Signals Metriken oder Traces fehlen, kann dies folgende Ursachen haben. Untersuchen Sie diese Ursachen nur, wenn Sie seit Ihrer letzten Aktualisierung bereits 15 Minuten darauf gewartet haben, dass Application Signals Daten erfasst und anzeigt.

- Stellen Sie sicher, dass die Bibliothek und das Framework, die Sie verwenden, vom ADOT-Java-Agenten unterstützt werden. Weitere Informationen finden Sie unter Bibliotheken/Frameworks.
- Stellen Sie sicher, dass der CloudWatch Agent läuft. Überprüfen Sie zunächst den Status der CloudWatch Agenten-Pods und stellen Sie sicher, dass sie alle im `Running` Status sind.

  ```
  kubectl -n amazon-cloudwatch get pods.
  ```

  Fügen Sie der CloudWatch Agent-Konfigurationsdatei Folgendes hinzu, um Debugging-Protokolle zu aktivieren, und starten Sie den Agenten anschließend neu.

  ```
  "agent": {
  >>>>>>> streams
    "region": "${REGION}",
    "debug": true
  },
  ```

  Suchen Sie dann in den CloudWatch Agent-Pods nach Fehlern.
- Suchen Sie nach Konfigurationsproblemen mit dem CloudWatch Agenten. Vergewissern Sie sich, dass sich Folgendes immer noch in der CloudWatch Agenten-Konfigurationsdatei befindet und der Agent seit dem Hinzufügen neu gestartet wurde.

  ```
  "agent": {
    "region": "${REGION}",
    "debug": true
  },
  ```

  Überprüfen Sie dann die OpenTelemetry Debugging-Protokolle auf Fehlermeldungen wie. `ERROR io.opentelemetry.exporter.internal.grpc.OkHttpGrpcExporter - Failed to export ...` Diese Meldungen könnten auf das Problem hinweisen.

  Wenn das Problem dadurch nicht behoben wird, speichern und überprüfen Sie die Umgebungsvariablen, deren Namen mit `OTEL_` beginnen, indem Sie den Pod mit dem `kubectl describe pod`-Befehl beschreiben.
- Um die OpenTelemetry Python-Debug-Protokollierung zu aktivieren, setzen Sie die Umgebungsvariable `OTEL_PYTHON_LOG_LEVEL` auf `debug` und stellen Sie die Anwendung erneut bereit.

- Prüfen Sie, ob die Berechtigungen für den Export von Daten aus dem Agenten falsch oder unzureichend sind. CloudWatch Wenn Sie `Access Denied` Nachrichten in den CloudWatch Agentenprotokollen sehen, könnte dies das Problem sein. Es ist möglich, dass die bei der Installation des CloudWatch Agenten geltenden Berechtigungen später geändert oder aufgehoben wurden.
- Achten Sie bei der Generierung von Telemetriedaten auf ein Problem mit AWS Distro for OpenTelemetry (ADOT).

  Stellen Sie sicher, dass die Anmerkungen zur Instrumentierung `instrumentation.opentelemetry.io/inject-java` und `sidecar.opentelemetry.io/inject-java` auf die Anwendungsbereitstellung angewendet werden, und dass der Wert `true` lautet. Ohne diese werden die Anwendungs-Pods nicht instrumentiert, selbst wenn das ADOT-Add-On korrekt installiert ist.

  Prüfen Sie als Nächstes, ob der `Init`-Container auf die Anwendung angewendet wurde und ob der `Ready`-Status `True` ist. Wenn der `init`-Container nicht bereit ist, sehen Sie sich den Status an, um den Grund zu erfahren.

  Wenn das Problem weiterhin besteht, gehen Sie wie folgt vor, um die Debug-Protokollierung im Java SDK zu aktivieren. OpenTelemetry Suchen Sie dann nach Nachrichten, die mit `ERROR` `io.telemetry` beginnen.

  Um die Debugging-Protokollierung zu aktivieren, setzen Sie die Umgebungsvariable `OTEL_JAVAAGENT_DEBUG` auf wahr und stellen Sie die Anwendung erneut bereit.
- Der Metrik-/Span-Exporter verwirft möglicherweise Daten. Um das herauszufinden, suchen Sie im Anwendungsprotokoll nach Meldungen, die `Failed to export...` beinhalten.
- Der CloudWatch Agent wird möglicherweise gedrosselt, wenn er Metriken oder Spans an Application Signals sendet. Suchen Sie in den Agentenprotokollen nach Meldungen, die auf eine Drosselung hinweisen. CloudWatch

Abhängigkeitsmetriken haben unbekannte Werte

Wenn Sie in den Dashboards von Application Signals UnknownOperationUnknownRemoteService, oder UnknownRemoteOperationfür einen Abhängigkeitsnamen oder einen Vorgang sehen, überprüfen Sie, ob das Auftreten von Datenpunkten für den unbekannten Remote-Service und den unbekannten Remote-Vorgang mit deren Bereitstellung übereinstimmt. Es handelt sich um ein bekanntes Problem bei Application Signals, das in einer zukünftigen Version behoben werden soll.

Umgang mit einem ConfigurationConflict bei der Verwaltung des Amazon CloudWatch Observability EKS-Add-ons

Wenn Sie bei der Installation oder Aktualisierung des Amazon CloudWatch Observability EKS-Add-ons einen Fehler feststellen, `Health Issue` der durch einen Typ `ConfigurationConflict` mit einer Beschreibung verursacht wird, die mit beginnt`Conflicts found when trying to apply. Will not continue due to resolve conflicts mode`, liegt das wahrscheinlich daran, dass Sie den CloudWatch Agenten und die zugehörigen Komponenten wie den ServiceAccount, den ClusterRole und den bereits auf dem Cluster ClusterRoleBinding installiert haben. Wenn das Add-on versucht, den CloudWatch Agenten und die zugehörigen Komponenten zu installieren und eine Änderung des Inhalts feststellt, schlägt es standardmäßig die Installation oder Aktualisierung fehl, um zu verhindern, dass der Status der Ressourcen auf dem Cluster überschrieben wird.

Wenn Sie versuchen, das Amazon CloudWatch Observability EKS-Add-on zu integrieren und dieser Fehler auftritt, empfehlen wir, ein vorhandenes CloudWatch Agenten-Setup zu löschen, das Sie zuvor auf dem Cluster installiert hatten, und dann das EKS-Add-on zu installieren. Stellen Sie sicher, dass Sie alle Anpassungen, die Sie möglicherweise am ursprünglichen CloudWatch Agenten-Setup vorgenommen haben, wie z. B. eine benutzerdefinierte Agentenkonfiguration, sichern und diese dem Amazon CloudWatch Observability EKS-Add-on zur Verfügung stellen, wenn Sie es das nächste Mal installieren oder aktualisieren. Wenn Sie den CloudWatch Agenten für das Onboarding in Container Insights bereits installiert hatten, finden Sie Den CloudWatch Agenten und Fluent Bit für Container Insights löschen weitere Informationen unter.

Alternativ unterstützt das Add-On eine Konfigurationsoption zur Konfliktlösung, die `OVERWRITE` spezifizieren kann. Sie können diese Option verwenden, um mit der Installation oder Aktualisierung des Add-Ons fortzufahren, indem Sie die Konflikte auf dem Cluster überschreiben. Wenn Sie die Amazon-EKS-Konsole verwenden, finden Sie die Methode zur Konfliktlösung, wenn Sie beim Erstellen oder Aktualisieren des Add-Ons die optionalen Konfigurationseinstellungen auswählen. Wenn Sie den verwenden AWS CLI, können Sie den in Ihren Befehl eingeben, `--resolve-conflicts OVERWRITE` um das Add-on zu erstellen oder zu aktualisieren.

Konfigurieren von Application Signals

⚠ Application Signals befindet sich in der Vorschauversion. Wenn Sie Feedback zu dieser Funktion haben, können Sie uns unter app-signals-feedback@amazon .com kontaktieren.

Dieser Abschnitt enthält Informationen zur Konfiguration von CloudWatch Anwendungssignalen.

Trace-Sampling-Rate

Wenn Sie Application Signals X-Ray aktivieren, ist das zentrale Sampling standardmäßig mit den Standardeinstellungen für die Sampling-Rate von `reservoir=1/s` und `fixed_rate=5%` aktiviert. Die Umgebungsvariablen für den AWS Distro for OpenTelemetry (ADOT) SDK-Agenten sind wie folgt festgelegt.

Umgebungsvariable	Wert	Hinweis
`OTEL_TRACES_SAMPLER`	`xray`	
`OTEL_TRACES_SAMPLER_ARG`	`endpoint=http://cloudwatch-agent.amazon-cloudwatch:2000`	Endpunkt des Agenten CloudWatch

Informationen zum Ändern der Sampling-Konfiguration finden Sie unter:

- Informationen zum Ändern des X-Ray-Sampling finden Sie unter Anpassen der Sampling-Regeln.
- Informationen zum Ändern der ADOT-Probenahme finden Sie unter Konfiguration des OpenTelemetry Collectors für die Röntgenfernabtastung

Wenn Sie das zentrale X-Ray-Sampling deaktivieren und stattdessen das lokale Sampling verwenden möchten, legen Sie die folgenden Werte für den ADOT-SDK-Java-Agenten wie folgt fest. Im folgenden Beispiel wird die Sampling-Rate auf 5 % festgelegt.

Umgebungsvariable	Wert
`OTEL_TRACES_SAMPLER`	`parentbased_traceidratio`
`OTEL_TRACES_SAMPLER_ARG`	`0.05`

Informationen zu erweiterten Sampling-Einstellungen finden Sie unter OTEL_TRACES_SAMPLER.

Verwalten Sie Operationen mit hoher Kardinalität

Application Signals enthält Einstellungen im CloudWatch Agenten, mit denen Sie die Kardinalität Ihrer Operationen und den Export von Kennzahlen verwalten können, um die Kosten zu optimieren. Standardmäßig wird die Metrikbegrenzungsfunktion aktiv, wenn die Anzahl der einzelnen Operationen für einen Service im Laufe der Zeit den Standardschwellenwert von 500 überschreitet. Sie können das Verhalten anpassen, indem Sie die Konfigurationseinstellungen anpassen.

Stellen Sie fest, ob die metrische Begrenzung aktiviert ist

Sie können die folgenden Methoden verwenden, um herauszufinden, ob die standardmäßige metrische Limitierung stattfindet. Ist dies der Fall, sollten Sie erwägen, die Kardinalitätssteuerung zu optimieren, indem Sie die Schritte im nächsten Abschnitt befolgen.

- Wählen Sie in der CloudWatch Konsole Application Signals, Services aus. Wenn Sie einen Vorgang namens AllOtherOperationsoder einen RemoteOperationbenannten Vorgang sehen AllOtherRemoteOperations, liegt eine Metrikbegrenzung vor.
- Wenn von Application Signals gesammelte Metriken den Wert `AllOtherOperations` für ihre `Operation` Dimension haben, erfolgt eine Metrikbegrenzung.
- Wenn von Application Signals gesammelte Metriken den Wert `AllOtherRemoteOperations` für ihre `RemoteOperation` Dimension haben, erfolgt eine Metrikbegrenzung.

Optimieren Sie die Kardinalitätskontrolle

Um Ihre Kardinalitätskontrolle zu optimieren, können Sie wie folgt vorgehen:

- Erstellen Sie benutzerdefinierte Regeln, um Operationen zu aggregieren.
- Konfigurieren Sie Ihre Richtlinie zur Begrenzung von Metriken.

Erstellen Sie benutzerdefinierte Regeln, um Operationen zu aggregieren

Operationen mit hoher Kardinalität können manchmal durch unangemessene Einzelwerte verursacht werden, die aus dem Kontext extrahiert wurden. Beispielsweise kann das Senden von HTTP/S-Anfragen, die Benutzer-IDs oder Sitzungs-IDs im Pfad enthalten, zu Hunderten von unterschiedlichen Vorgängen führen. Um solche Probleme zu lösen, empfehlen wir, den CloudWatch Agenten mit Anpassungsregeln zu konfigurieren, um diese Vorgänge neu zu schreiben.

In Fällen, in denen die Generierung zahlreicher verschiedener Messwerte durch einzelne `RemoteOperation` Aufrufe, wie z. B., und ähnliche Anfragen `PUT /api/customer/owners/123PUT /api/customer/owners/456`, stark zunimmt, empfehlen wir, diese Vorgänge in einem einzigen `RemoteOperation` Vorgang zu konsolidieren. Ein Ansatz besteht insbesondere `PUT /api/customer/owners/{ownerId}` darin, alle `RemoteOperation` Anrufe, die mit beginnen, `PUT /api/customer/owners/` auf ein einheitliches Format zu standardisieren. Das folgende Beispiel illustriert dies. Informationen zu anderen Anpassungsregeln finden Sie unter CloudWatch Anwendungssignale aktivieren.

```
{
   "logs":{
      "metrics_collected":{
         "app_signals":{
            "rules":[
               {
                  "selectors":[
                     {
                        "dimension":"RemoteOperation",
                        "match":"PUT /api/customer/owners/*"
                     }
                  ],
                  "replacements":[
                     {
                        "target_dimension":"RemoteOperation",
                        "value":"PUT /api/customer/owners/{ownerId}"
                     }
                  ],
                  "action":"replace"
               }
            ]
         }
      }
   }
}
```

In anderen Fällen wurden Metriken mit hoher Kardinalität möglicherweise zu aggregiert`AllOtherRemoteOperations`, und es ist möglicherweise unklar, welche spezifischen Metriken enthalten sind. Der CloudWatch Agent ist in der Lage, die unterbrochenen Operationen zu protokollieren. Um unterbrochene Operationen zu identifizieren, verwenden Sie die Konfiguration im folgenden Beispiel, um die Protokollierung zu aktivieren, bis das Problem erneut auftritt. Untersuchen Sie dann die CloudWatch Agentenprotokolle (auf die über Container `stdout` - oder

EC2-Protokolldateien zugegriffen werden kann) und suchen Sie nach dem Schlüsselwort. `drop metric data`

```
{
  "agent": {
    "config": {
      "agent": {
        "debug": true
      },
      "traces": {
        "traces_collected": {
          "app_signals": {
          }
        }
      },
      "logs": {
        "metrics_collected": {
          "app_signals": {
            "limiter": {
              "log_dropped_metrics": true
            }
          }
        }
      }
    }
  }
}
```

Erstellen Sie Ihre Richtlinie zur Begrenzung von Metriken

Wenn die Standardkonfiguration für metrische Beschränkungen die Kardinalität für Ihren Service nicht berücksichtigt, können Sie die Konfiguration der metrischen Limitierung anpassen. Fügen Sie dazu unter dem `limiter` Abschnitt in der CloudWatch Agent-Konfigurationsdatei einen `logs/metrics_collected/app_signals` Abschnitt hinzu.

Im folgenden Beispiel wird der Schwellenwert für die Metrikbegrenzung von 500 verschiedenen Metriken auf 100 gesenkt.

```
{
  "logs": {
    "metrics_collected": {
      "app_signals": {
```

```
        "limiter": {
          "drop_threshold": 100
        }
      }
    }
  }
}
```

Servicelevel-Ziele (SLOs)

⚠ Application Signals befindet sich in der Vorschauversion. Wenn Sie Feedback zu dieser Funktion haben, können Sie uns unter app-signals-feedback@amazon .com kontaktieren.

Sie können Application Signals verwenden, um Servicelevel-Ziele für die Services für Ihre kritischen Geschäftsabläufe zu erstellen. Wenn Sie SLOs für diese Dienste erstellen, können Sie sie im SLO-Dashboard verfolgen, sodass Sie einen at-a-glance Überblick über Ihre wichtigsten Abläufe haben.

Neben der Erstellung einer Schnellansicht, in der sich Ihre Mitarbeiter über den aktuellen Status kritischer Vorgänge informieren können, können Sie mit Hilfe von SLOs auch die längerfristige Leistung Ihrer Services verfolgen, um sicherzustellen, dass sie Ihren Erwartungen entsprechen. Wenn Sie Service Level Agreements mit Kunden abgeschlossen haben, sind SLOs ein hervorragendes Instrument, um sicherzustellen, dass diese eingehalten werden.

Die Bewertung des Zustands Ihrer Services mithilfe von SLOs beginnt mit der Festlegung klarer, messbarer Ziele auf der Grundlage wichtiger Leistungsmetriken – Servicelevel-Indikator (SLIs). Mit einem SLO wird die SLI-Leistung anhand des von Ihnen festgelegten Schwellenwerts und Ziels verglichen und es wird gemeldet, wie weit oder wie nahe Ihre Anwendungsleistung am Schwellenwert liegt.

Application Signals hilft Ihnen dabei, SLOs für Ihre wichtigsten Leistungsmetriken festzulegen. Application Signals erfasst automatisch `Latency`- und `Availability`-Metriken für jeden Service und Vorgang, den es entdeckt, und diese Metriken eignen sich oft ideal für die Verwendung als SLIs. Mit dem Assistenten zur SLO-Erstellung können Sie diese Metriken für Ihre SLOs verwenden. Anschließend können Sie den Status all Ihrer SLOs mit den Dashboards von Application Signals verfolgen.

Sie können SLOs für bestimmte Vorgänge einrichten, die Ihr Service aufruft oder verwendet. Zusätzlich zu den Metriken und können Sie jede beliebige CloudWatch Metrik oder jeden metrischen Ausdruck als SLI verwenden. `Latency` `Availability`

Die Erstellung von SLOs ist sehr wichtig, um den größtmöglichen Nutzen aus CloudWatch Application Signals zu ziehen. Nachdem Sie SLOs erstellt haben, können Sie ihren Status in der Application Signals Console einsehen, um schnell zu sehen, welche Ihrer kritischen Services und Vorgänge gut funktionieren und welche nicht. Die Tatsache, dass SLOs nachverfolgt werden können, bietet die folgenden großen Vorteile:

- Es ist für Ihre Servicebetreiber einfacher, den aktuellen Betriebsstatus kritischer Services, gemessen am SLI, zu ermitteln. Auf diese Weise können sie schnell fehlerhafte Services und Betriebsabläufe untersuchen und identifizieren.
- Sie können Ihre Serviceleistung anhand messbarer Geschäftsziele über längere Zeiträume hinweg verfolgen.

Indem Sie entscheiden, worauf Sie SLOs setzen möchten, priorisieren Sie das, was für Sie wichtig ist. Die Dashboards von Application Signals enthalten automatisch Informationen darüber, was Sie priorisiert haben.

Wenn Sie ein SLO erstellen, können Sie sich auch dafür entscheiden, gleichzeitig CloudWatch Alarme zu erstellen, um die SLOs zu überwachen. Sie können Alarme einrichten, die sowohl auf Überschreitungen des Schwellenwerts als auch auf Warnstufen achten. Diese Alarme können Sie automatisch benachrichtigen, wenn die SLO-Metriken den von Ihnen festgelegten Schwellenwert überschreiten oder wenn sie sich einem Warnschwellenwert nähern. Wenn sich ein SLO beispielsweise seinem Warnschwellenwert nähert, können Sie darüber informiert werden, dass Ihr Team möglicherweise die Kundenabwanderung in der Anwendung verlangsamen muss, um sicherzustellen, dass die langfristigen Leistungsziele erreicht werden.

Themen

SLO-Konzepte

Ein SLO-Konzept enthält die folgenden Komponenten:

- Ein Servicelevel-Indikator (SLI), bei dem es sich um eine wichtige Leistungsmetrik handelt, die Sie angeben. Der SLI stellt das gewünschte Leistungsniveau für Ihre Anwendung dar. Application Signals erfasst automatisch die wichtigen `Latency`- und `Availability`-Metriken für jeden Service und Vorgang, den es entdeckt, und diese Metriken eignen sich oft ideal für die Verwendung mit SLOs.

  Sie wählen den Schwellenwert, den Sie für Ihr SLI verwenden möchten. Zum Beispiel 200 ms für die Latenz.
- Ein Ziel oder Erreichungsziel. Dabei handelt es sich um den Prozentsatz der Zeit, in der der SLI den Schwellenwert voraussichtlich in jedem Zeitintervall erreicht. Die Zeitintervalle können so kurz wie Stunden oder so lang wie ein Jahr sein.

  Intervalle können entweder Kalenderintervalle oder fortlaufende Intervalle sein.
  - Kalenderintervalle werden auf den Kalender abgestimmt, z. B. ein SLO, das pro Monat erfasst wird. CloudWatch passt die Zahlen für Gesundheit, Budget und Leistungsstand automatisch an die Anzahl der Tage in einem Monat an. Kalenderintervalle eignen sich besser für Geschäftsziele, die kalendergerecht gemessen werden.
  - Rollende Intervalle werden fortlaufend berechnet. Rollende Intervalle eignen sich besser, um das aktuelle Benutzererlebnis Ihrer Anwendung nachzuverfolgen.
- Der Zeitraum ist kürzer, und viele Zeiträume bilden ein Intervall. Die Leistung der Anwendung wird in jedem Zeitraum innerhalb des Intervalls mit der SLI verglichen. Für jeden Zeitraum wird festgestellt, ob die Anwendung entweder die erforderliche Leistung erreicht hat oder nicht.

Ein Ziel von 99 % bei einem Kalenderintervall von einem Tag und einem Zeitraum von 1 Minute bedeutet beispielsweise, dass die Anwendung die Erfolgsschwelle in 99 % der Zeiträume von 1 Minute am Tag erreichen oder erreichen muss. Ist dies der Fall, ist der SLO für diesen Tag erfüllt. Der nächste Tag ist ein neues Bewertungsintervall, und die Anwendung muss während 99 % der Zeiträume von 1 Minute am zweiten Tag die Erfolgsschwelle erreichen oder erreichen, um den SLO für diesen zweiten Tag zu erfüllen.

Ein SLI kann auf einer der neuen Standard-Anwendungsmetriken basieren, die von Application Signals erfasst wurden. Alternativ kann es sich um eine beliebige CloudWatch Metrik oder einen beliebigen metrischen Ausdruck handeln. Die Standard-Anwendungsmetriken, die Sie für eine SLI

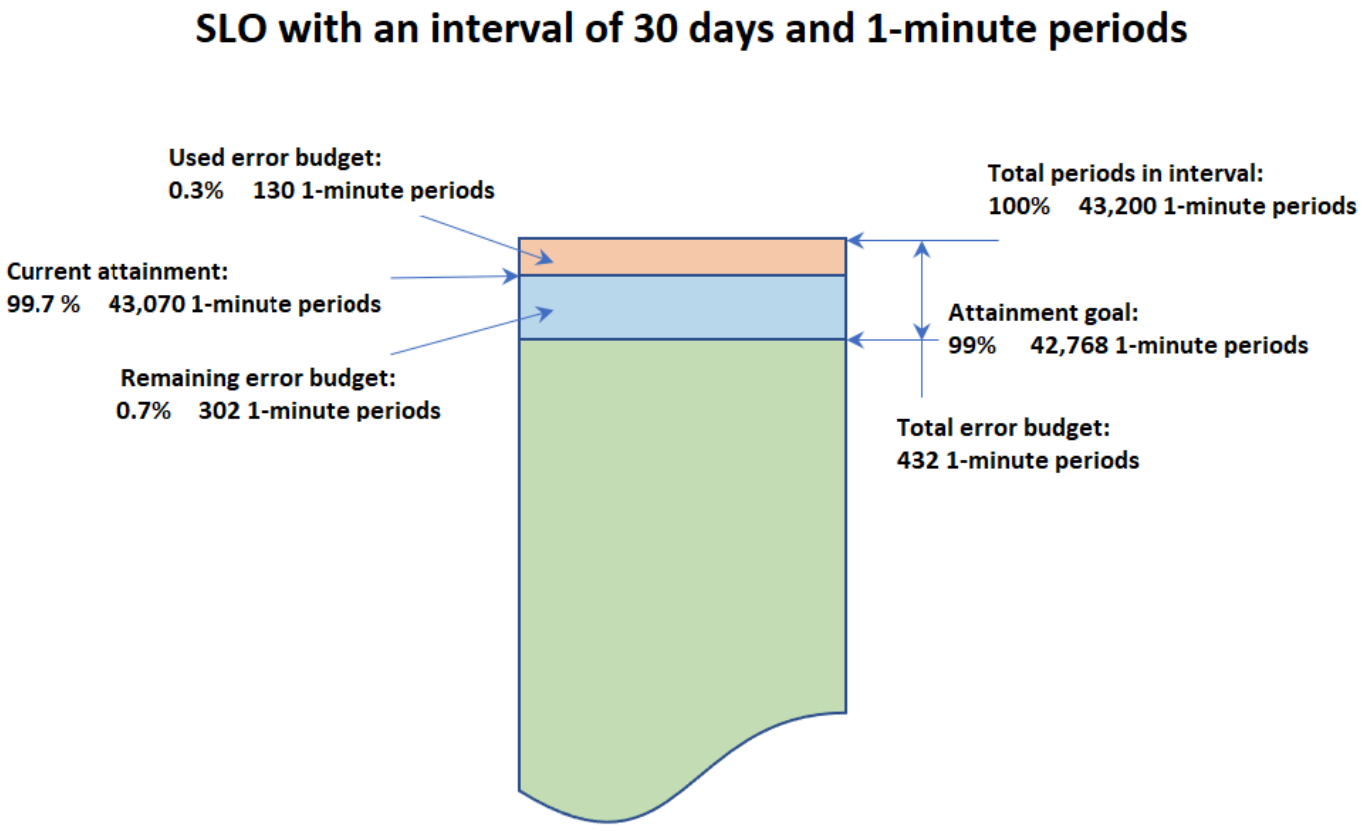
verwenden können, sind `Latency` und `Availability`. `Availability` stellt die erfolgreichen Antworten geteilt durch die Gesamtzahl der Anfragen dar. Sie wird als (1 - Störungsrate)*100 berechnet, wobei es sich bei Fehlerantworten um `5xx`-Fehler handelt. Erfolgsantworten sind Antworten ohne 5XX-Fehler. 4XX-Antworten werden als erfolgreich behandelt.

> **Note**
>
> Derzeit werden nur zeitraumbasierte Berechnungen unterstützt. Unterstützung für volumen- oder anforderungsbasierten Berechnungen ist für zukünftige Versionen geplant.

Fehlerbudget und Erreichung berechnen

Wenn Sie Informationen zu einem SLO anzeigen, sehen Sie dessen aktuellen Zustand und sein Fehlerbudget. Das Fehlerbudget gibt an, wie lange innerhalb des Intervalls der Schwellenwert überschritten werden kann, wobei der SLO aber trotzdem eingehalten werden kann. Das Gesamtfehlerbudget ist die Gesamtdauer der Sicherheitsverletzung, die während des gesamten Intervalls toleriert werden kann. Das verbleibende Fehlerbudget ist die verbleibende Dauer der Sicherheitsverletzung, die im aktuellen Intervall toleriert werden kann. Dies ist der Fall, nachdem die bereits eingetretene Zeit für Verstöße vom Gesamtfehlerbudget abgezogen wurde.

Die folgende Abbildung veranschaulicht die Konzepte für das Erreichungs- und das Fehlerbudget für ein Ziel mit einem Intervall von 30 Tagen, Zeiträumen von 1 Minute und einer Zielerreichung von 99 %. 30 Tage beinhalten 43 200 Zeiträume von einer Minute. 99 % von 43 200 entsprechen 42 768 Minuten im Monat, sodass 42 768 Minuten im Monat einwandfrei sein müssen, damit der SLO eingehalten werden kann. Bisher waren 130 der 1-Minuten-Zeiträume im aktuellen Intervall fehlerbehaftet.



Den Erfolg innerhalb der einzelnen Zeiträumen ermitteln

Innerhalb jedes Zeitraums werden die SLI-Daten auf der Grundlage der für den SLI verwendeten Statistik zu einem einzigen Datenpunkt zusammengefasst. Dieser Datenpunkt stellt die gesamte Länge des Zeitraums dar. Dieser einzelne Datenpunkt wird mit dem SLI-Schwellenwert verglichen, um festzustellen, ob der Zeitraum fehlerfrei ist. Wenn Sie auf dem Dashboard fehlerhafte Perioden während des aktuellen Zeitraums sehen, können Ihre Servicemitarbeiter darauf aufmerksam gemacht werden, dass der Service untersucht werden muss.

Wenn festgestellt wird, dass der Zeitraum fehlerhaft ist, wird die gesamte Dauer des Zeitraums im Fehlerbudget als fehlerhaft gewertet. Wenn Sie das Fehlerbudget verfolgen, können Sie feststellen, ob der Service über einen längeren Zeitraum die von Ihnen gewünschte Leistung erzielt.

Ein SLO erstellen

Wir empfehlen, dass Sie für Ihre kritischen Anwendungen sowohl Latenz- als auch Verfügbarkeits-SLOs festlegen. Diese von Application Signals gesammelten Metriken entsprechen den gemeinsamen Geschäftszielen.

Sie können SLOs auch für jede CloudWatch Metrik oder jeden metrischen mathematischen Ausdruck festlegen, der zu einer einzigen Zeitreihe führt.

Wenn Sie zum ersten Mal ein SLO in Ihrem Konto erstellen, CloudWatch wird automatisch die AWSServiceRoleForCloudWatchApplicationSignalsserviceverknüpfte Rolle in Ihrem Konto erstellt, sofern sie noch nicht vorhanden ist. Diese dienstbezogene Rolle ermöglicht CloudWatch das Sammeln von CloudWatch Logdaten, X-Ray-Trace-Daten, CloudWatch Metrikdaten und Tagging-Daten von Anwendungen in Ihrem Konto. Weitere Informationen zu CloudWatch dienstbezogenen Rollen finden Sie unter. Verwenden von serviceverknüpften Rollen für CloudWatch

So erstellen Sie ein SLO

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Servicelevel-Ziele (SLO).
3. Wählen Sie SLO erstellen.
4. Geben Sie einen Namen für das SLO ein. Wenn Sie den Namen eines Services oder Vorgangs zusammen mit entsprechenden Schlüsselwörtern wie Latenz oder Verfügbarkeit angeben, können Sie bei der Untersuchung schnell erkennen, was der SLO-Status bedeutet.
5. Führen Sie für Servicelevel-Indikator (SLI) festlegen einen der folgenden Schritte aus:
   - Um das SLO auf eine der Standard-Anwendungsmetriken `Latency` oder `Availability` festzulegen:
     a. Wählen Sie Service-Vorgang.
     b. Wählen Sie den Service aus, den dieses SLO überwachen soll.
     c. Wählen Sie den Vorgang aus, den dieses SLO überwachen soll.

        Die Dropdown-Menüs Service auswählen und Vorgang auswählen werden mit Services und Vorgängen gefüllt, die in den letzten 24 Stunden aktiv waren.
     d. Wählen Sie entweder Verfügbarkeit oder Latenz und legen Sie dann den Schwellenwert fest.
   - Um den SLO für eine beliebige CloudWatch Metrik oder einen CloudWatch metrischen mathematischen Ausdruck festzulegen:
     a. Wählen Sie CloudWatch Metrisch.
     b. Wählen Sie CloudWatch Metrik auswählen.

Der Bildschirm Metrik auswählen wird angezeigt. Verwenden Sie die Registerkarten Durchsuchen oder Abfragen, um die gewünschte Metrik zu finden, oder erstellen Sie einen mathematischen Ausdruck für die Metrik.

Nachdem Sie die gewünschte Metrik ausgewählt haben, wählen Sie die Registerkarte Graphische Metriken und dann die Statistik und den Zeitraum aus, die für das SLO verwendet werden sollen. Wählen Sie dann Select Metric (Metrik auswählen) aus.

Weitere Informationen zu diesen Bildschirmen finden Sie unter Grafisches Darstellen von Metriken und Fügen Sie einem CloudWatch Diagramm einen mathematischen Ausdruck hinzu.

c. Wählen Sie unter Bedingung festlegen einen Vergleichsoperator und einen Schwellenwert aus, den das SLO als Erfolgsindikator verwenden soll.

6. Wenn Sie in Schritt 5 Service-Vorgang ausgewählt haben, können Sie optional Zusätzliche Einstellungen auswählen und dann die Länge des Zeitraums für dieses SLO anpassen.
7. Legen Sie das Intervall und das Erreichungsziel für das SLO fest. Weitere Informationen zu Intervallen und Erreichungszielen sowie zu deren Zusammenspiel finden Sie unter SLO-Konzepte.
8. (Optional) Legen Sie einen oder mehrere CloudWatch Alarme oder einen Warnschwellenwert für den SLO fest.

   a. CloudWatch Alarme können Amazon SNS verwenden, um Sie proaktiv zu benachrichtigen, wenn eine Anwendung aufgrund ihrer SLI-Leistung fehlerhaft ist.

      Um einen Alarm zu erstellen, wählen Sie eines der Alarm-Kontrollkästchen aus und geben Sie das Amazon-SNS-Thema ein – oder erstellen Sie eines – welches für Benachrichtigungen verwendet werden soll, wenn der Alarm in den `ALARM`-Status wechselt. Weitere Informationen zu CloudWatch Alarmen finden Sie unter. CloudWatch Amazon-Alarme verwenden Für die Erstellung von Alarmen fallen Gebühren an. Weitere Informationen zur CloudWatch Preisgestaltung finden Sie unter CloudWatch Amazon-Preise.

   b. Wenn Sie einen Warnschwellenwert festlegen, wird dieser auf den Bildschirmen von Application Signals angezeigt und hilft Ihnen dabei, SLOs zu identifizieren, bei denen die Gefahr besteht, dass sie nicht erfüllt werden, auch wenn sie derzeit fehlerfrei sind.

Um einen Warnschwellenwert festzulegen, geben Sie den Schwellenwert im Feld Warnschwellenwert ein. Wenn das Fehlerbudget des SLO unter dem Warnschwellenwert liegt, wird das SLO auf mehreren Bildschirmen von Application Signals mit Warnung gekennzeichnet. Warnschwellenwerte werden auch in den Grafiken zum Fehlerbudget angezeigt. Sie können auch einen SLO-Warnalarm erstellen, der auf dem Warnschwellenwert basiert.

9. Um diesem SLO Tags hinzuzufügen, wählen Sie die Registerkarte Tags und dann Neues Tag hinzufügen. Mit Tags können Sie Ressourcen verwalten, identifizieren, organisieren, suchen und filtern. Weitere Informationen über das Markieren finden Sie unter Markieren Ihrer AWS -Ressourcen.

> **Note**
>
> Wenn die Anwendung, auf die sich dieses SLO bezieht, registriert ist AWS Service Catalog AppRegistry, können Sie das `awsApplication` Tag verwenden, um dieses SLO dieser Anwendung zuzuordnen AppRegistry. Weitere Informationen finden Sie unter Was ist AppRegistry?

10. Wählen Sie SLO erstellen. Wenn Sie sich außerdem dafür entscheiden, einen oder mehrere Alarme zu erstellen, ändert sich der Name der Schaltfläche entsprechend.

SLO-Status anzeigen und untersuchen

Mithilfe der Service Level Objectives oder der Services-Optionen in der CloudWatch Konsole können Sie sich schnell einen Überblick über den Zustand Ihrer SLOs verschaffen. Die Ansicht „Dienste“ bietet einen at-a-glance Überblick über das Verhältnis fehlerhafter Dienste, das auf der Grundlage der von Ihnen festgelegten SLOs berechnet wird. Weitere Informationen zur Verwendung der Services-Option finden Sie unter Den Betriebsstatus Ihrer Anwendungen mit Application Signals überwachen.

Die Ansicht Servicelevel-Ziele bietet eine übergeordnete Ansicht Ihrer Organisation. Sie können die erfüllten und nicht erfüllten SLOs als Ganzes sehen. Auf diese Weise erhalten Sie einen Überblick darüber, wie viele Ihrer Services und Abläufe gemäß den von Ihnen ausgewählten SLIs über längere Zeiträume Ihren Erwartungen entsprechen.

So zeigen Sie alle SLOs in der Servicelevel-Ziele-Ansicht an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

2. Wählen Sie im Navigationsbereich Servicelevel-Ziele (SLO).

   Die Liste der Servicelevel-Ziele (SLO) wird angezeigt.

   In der SLI-Status-Spalte können Sie schnell den aktuellen Status Ihrer SLOs einsehen. Um die SLOs so zu sortieren, dass alle fehlerhaften SLOs ganz oben in der Liste stehen, wählen Sie die SLI-Status-Spalte aus, bis alle fehlerhaften SLOs ganz oben stehen.

   Die SLO-Tabelle hat die folgenden standardmäßigen Spalten. Sie können anpassen, welche Spalten angezeigt werden, indem Sie das Zahnradsymbol über der Liste auswählen. Weitere Informationen zu Zielen, SLIs, erreichten Zielen und Intervallen finden Sie unter SLO-Konzepte.

   - Der Name des SLO.
   - In der Ziel-Spalte wird der Prozentsatz der Zeiträume in jedem Intervall angezeigt, bei denen der SLI-Schwellenwert erfolgreich erreicht werden muss, damit das SLO-Ziel erreicht wird. Außerdem wird die Intervall-Länge für das SLO angezeigt.
   - Der SLI-Status zeigt an, ob der aktuelle Betriebsstatus der Anwendung fehlerfrei ist oder nicht. Wenn ein Zeitraum innerhalb des aktuell ausgewählten Zeitraums für das SLO fehlerhaft war, wird der SLI-Status als Fehlerhaft angezeigt.
   - Das Endziel ist das Erreichungsniveau, das am Ende des ausgewählten Zeitraums erreicht wurde. Sortieren Sie nach dieser Spalte, um die SLOs zu finden, bei denen die Gefahr am größten ist, dass sie nicht eingehalten werden.
   - Das Erreichungs-Delta ist der Unterschied in der Leistungsstufe zwischen dem Beginn und dem Ende des ausgewählten Zeitraums. Ein negatives Delta bedeutet, dass die Metrik nach unten tendiert. Sortieren Sie nach dieser Spalte, um die neuesten Trends der SLOs zu sehen.
   - Das Budget für Endfehler (%) ist der Prozentsatz der Gesamtzeit in dem Zeitraum, in dem es zu fehlerhaften Zeiträumen kommen kann und das SLO trotzdem erfolgreich erreicht werden kann. Wenn Sie diesen Wert auf 5 % setzen und der SLI in 5 % oder weniger der verbleibenden Zeiträumen des Intervalls fehlerhaft ist, wird das SLO trotzdem erfolgreich erreicht.
   - Das Fehlerbudget-Delta ist die Differenz im Fehlerbudget zwischen dem Start und dem Ende des ausgewählten Zeitraums. Ein negatives Delta bedeutet, dass die Metrik nach unten tendiert.
   - Beim Endfehlerbudget (Zeit) handelt es sich um die tatsächliche Zeit innerhalb des Intervalls, die fehlerhaft sein kann, während das SLO trotzdem erfolgreich erreicht werden muss. Wenn

dieser Wert beispielsweise 14 Minuten beträgt und der SLI während des verbleibenden Intervalls weniger als 14 Minuten fehlerhaft ist, wird das SLO trotzdem erfolgreich erreicht.

- In den Spalten Service, Vorgang und Typ werden Informationen darüber angezeigt, für welchen Service und welchen Betrieb dieses SLO eingerichtet ist.

3. Aktivieren Sie das Optionsfeld neben dem SLO-Namen, um die Budgets für Erreichen und Fehler für ein SLO anzuzeigen.

   Die Grafiken oben auf der Seite zeigen den Budgetstatus des SLO-Erreichens und des Fehlerbudgets. Ein Diagramm über die SLI-Metrik, die diesem SLO zugeordnet ist, wird ebenfalls angezeigt.

4. Um ein SLO, das sein Ziel nicht erreicht, genauer zu untersuchen, wählen Sie den Service- oder Vorgangsnamen, der diesem SLO zugeordnet ist. Sie werden auf die Detailseite weitergeleitet, auf der Sie eine weitere Auswahl vornehmen können. Weitere Informationen finden Sie unter Auf der Seite mit den Servicedetails können Sie detaillierte Serviceaktivitäten und den Betriebsstatus anzeigen.

5. Um den Zeitraum der Diagramme und Tabellen auf der Seite zu ändern, wählen Sie oben auf dem Bildschirm einen neuen Zeitraum aus.

Ein vorhandenes SLO bearbeiten

Gehen Sie folgendermaßen vor, um eine bestehende SLO zu bearbeiten. Wenn Sie ein SLO bearbeiten, können Sie nur den Schwellenwert, das Intervall, das Erreichungsziel und die Tags ändern. Um andere Aspekte wie Service, Betrieb oder Metrik zu ändern, erstellen Sie ein neues SLO, anstatt ein vorhandenes zu bearbeiten.

Wenn Sie einen Teil einer SLO-Kernkonfiguration ändern, z. B. einen Zeitraum oder einen Schwellenwert, werden alle vorherigen Datenpunkte und Bewertungen in Bezug auf Leistung und Zustand ungültig. Das SLO wird effektiv gelöscht und neu erstellt.

> **Note**
>
> Wenn Sie ein SLO bearbeiten, werden die mit diesem SLO verknüpften Alarme nicht automatisch aktualisiert. Möglicherweise müssen Sie die Alarme aktualisieren, damit sie mit dem SLO synchron bleiben.

So bearbeiten Sie ein vorhandenes SLO

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Servicelevel-Ziele (SLO).
3. Aktivieren Sie das Optionsfeld neben dem SLO, das Sie bearbeiten möchten, und wählen Sie Aktionen, SLO bearbeiten aus.
4. Nehmen Sie die gewünschten Änderungen vor und wählen Sie dann Änderungen speichern.

Ein SLO löschen

Gehen Sie folgendermaßen vor, um ein bestehendes SLO zu löschen.

> **Note**
>
> Wenn Sie ein SLO löschen, werden die mit diesem SLO verknüpften Alarme nicht automatisch gelöscht. Sie müssen sie selbst löschen. Weitere Informationen finden Sie unter Verwalten von Alarmen.

So löschen Sie ein SLO

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Servicelevel-Ziele (SLO).
3. Aktivieren Sie das Optionsfeld neben dem SLO, das Sie bearbeiten möchten, und wählen Sie Aktionen, SLO löschen aus.
4. Wählen Sie Bestätigen aus.

Den Betriebsstatus Ihrer Anwendungen mit Application Signals überwachen

> **Application Signals befindet sich in der Vorschauversion. Wenn Sie Feedback zu dieser Funktion haben, können Sie uns unter app-signals-feedback@amazon .com kontaktieren.**

Verwenden Sie Application Signals in der CloudWatch Konsole, um den Betriebsstatus Ihrer Anwendungen zu überwachen und Fehler zu beheben:

- Ihre Anwendungsservices überwachen – Im Rahmen der täglichen Betriebsüberwachung finden Sie auf der Seite Services eine Zusammenfassung all Ihrer Services. Sehen Sie sich die Services mit der höchsten Störungsrate oder Latenz an und finden Sie heraus, welche Services fehlerhafte Servicelevel-Indikatoren (SLIs) aufweisen. Wählen Sie einen Service aus, um die Service-Detailseite zu öffnen und detaillierte Metriken, Servicebetriebe, Synthetics-Canarys und Kundenanfragen anzuzeigen. Auf diese Weise können Sie die Ursache betrieblicher Probleme ermitteln und beheben.
- Ihre Anwendungstopologie untersuchen – Verwenden Sie die Service-Karte, um Ihre Anwendungstopologie im Laufe der Zeit zu verstehen und zu überwachen, einschließlich der Beziehungen zwischen Clients, Synthetics-Canarys, Services und Abhängigkeiten. Sehen Sie sich sofort den Status des Servicelevel-Indikators (SLI) an und lassen Sie sich wichtige Kennzahlen wie Aufrufvolumen, Störungsrate und Latenz anzeigen. Detailliertere Informationen finden Sie auf der Seite mit den Servicedetails.

Sehen Sie sich ein Beispielszenario an, das zeigt, wie diese Seiten zur schnellen Behebung eines Betriebszustands eines Services verwendet werden können, von der ersten Erkennung bis zur Identifizierung der Grundursache.

Wie Application Signals die Überwachung des Betriebsstatus ermöglicht

Nachdem Sie Ihre Anwendung für Application Signals aktiviert haben, werden Ihre Anwendungsservices, APIs und deren Abhängigkeiten automatisch erkannt und auf den Seiten Services, Servicedetails und Service-Karte angezeigt. Application Signals sammelt Informationen aus verschiedenen Quellen, um die Serviceerkennung und die Überwachung des Betriebsstatus zu ermöglichen:

- AWS Distro for OpenTelemetry (ADOT) — Im Rahmen der Aktivierung von Application Signals wird eine OpenTelemetry Java-Bibliothek für automatische Instrumentierung so konfiguriert, dass sie Metriken und Traces ausgibt, die vom Agenten gesammelt werden. CloudWatch Die Metriken und Traces werden verwendet, um die Erkennung von Services, Vorgängen, Abhängigkeiten und anderen Serviceinformationen zu ermöglichen.
- Servicelevel-Ziele (SLOs) – Nachdem Sie Servicelevel-Ziele für Ihre Services erstellt haben, wird auf den Seiten Services, Servicedetails und Service-Karte der Zustand des Servicelevel-Indikators (SLI) angezeigt. SLIs können Latenz, Verfügbarkeit und andere Betriebsmetriken überwachen.

- CloudWatch Synthetics Canaries — Wenn Sie X-Ray Tracing auf Ihren Canaries konfigurieren, werden Aufrufe Ihrer Dienste von Ihren Canary-Skripten aus mit Ihrem Dienst verknüpft und auf der Service-Detailseite angezeigt.
- CloudWatch Real User Monitoring (RUM) — Wenn X-Ray Tracing auf Ihrem CloudWatch RUM-Webclient aktiviert ist, werden Anfragen an Ihre Dienste automatisch verknüpft und auf der Service-Detailseite angezeigt.
- AWS Service Catalog AppRegistry— Application Signals erkennt automatisch AWS Ressourcen in Ihrem Konto und ermöglicht es Ihnen, sie in logischen Anwendungen zu gruppieren, die in erstellt wurden. AppRegistry Der auf der Services-Seite angezeigte Anwendungsname basiert auf der zugrunde liegenden Rechenressource, auf der Ihre Services ausgeführt werden.

> **Note**
>
> Application Signals zeigt Ihre Services und Operationen auf der Grundlage von Metriken und Traces an, die innerhalb des von Ihnen ausgewählten aktuellen Zeitfilters ausgegeben wurden. (Standardmäßig sind dies die letzten drei Stunden.) Wenn im aktuellen Zeitfilter für einen Services, einen Vorgang, eine Abhängigkeit, einen Synthetics-Canary oder eine Client-Seite keine Aktivität vorhanden ist, wird diese nicht angezeigt.
> Derzeit können bis zu 1 000 Services angezeigt werden. Die Erkennung Ihrer Services und der Servicetopologie kann sich um bis zu 10 Minuten verzögern. Die Bewertung des Zustands Ihres Servicelevel-Indikators (SLI) kann sich um bis zu 15 Minuten verzögern.

Allgemeine Serviceaktivität und den Betriebsstatus auf der Services-Seite anzeigen

> **Application Signals befindet sich in der Vorschauversion für Amazon CloudWatch und kann sich ändern.**

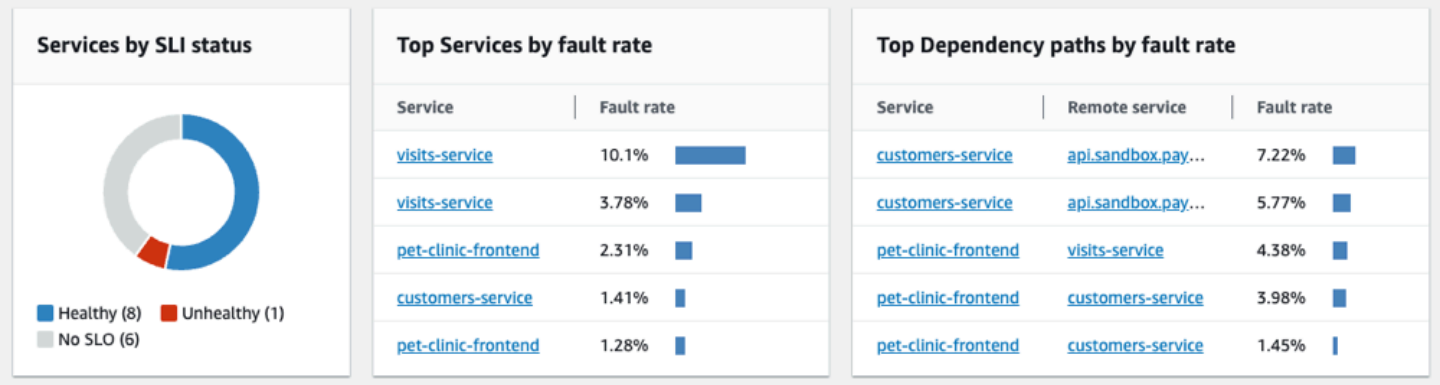
Auf der Services-Seite finden Sie eine Liste Ihrer Services, die für Application Signals aktiviert sind. Sie können auch Betriebsmetriken einsehen und schnell feststellen, welche Services fehlerhafte Servicelevel-Indikatoren (SLIs) aufweisen. Gehen Sie ins Deital, um nach Leistungsanomalien zu suchen, und ermitteln Sie die Ursache betrieblicher Probleme. Um diese Seite aufzurufen, öffnen

Sie die CloudWatch Konsole und wählen Sie im linken Navigationsbereich im Abschnitt Application Signals die Option Services aus.

Sich über Betriebsstatus-Metriken für Ihre Services informieren

Oben auf der Services-Seite finden Sie ein Diagramm zum allgemeinen Betriebsstatus des Services und mehrere Tabellen, in denen die wichtigsten Services und Serviceabhängigkeiten nach Fehlerrate angezeigt werden. Das Services-Diagramm auf der linken Seite zeigt eine Aufschlüsselung der Anzahl der Services, die während des aktuellen Zeitfilters auf Seitenebene fehlerfreie oder fehlerhafte Servicelevel-Indikatoren (SLIs) aufwiesen. SLIs können Latenz, Verfügbarkeit und andere Betriebsmetriken überwachen.

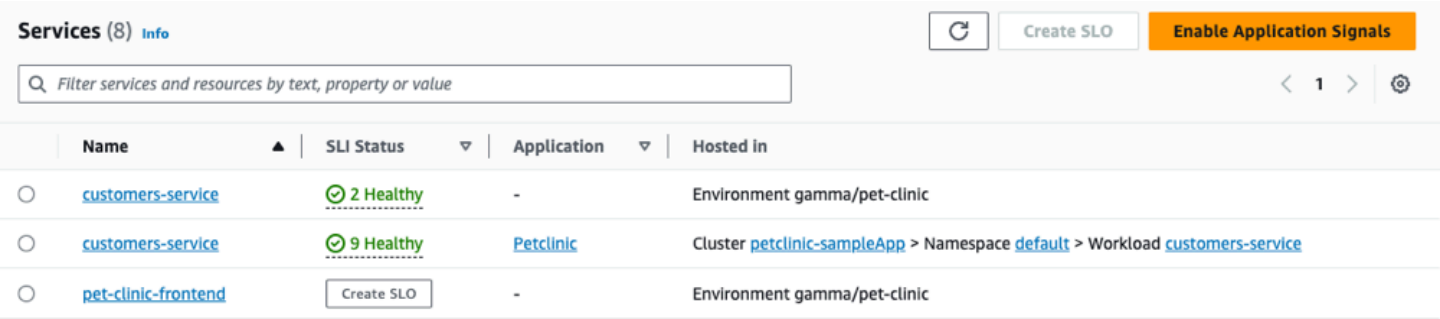
Die beiden Tabellen neben dem Diagramm zeigen eine Liste der wichtigsten Services nach Fehlerrate. Wählen Sie in einer der Tabellen einen beliebigen Service-Namen aus, um eine Seite mit den Service-Details zu öffnen und detaillierte Informationen zum den Service-Vorgängen anzuzeigen. Wählen Sie einen Abhängigkeitspfad, um die Detailseite zu öffnen und Informationen zu den Service-Abhängigkeiten anzuzeigen. In beiden Tabellen werden Informationen für die letzten drei Stunden angezeigt, auch wenn oben rechts auf der Seite ein Filter für längere Zeiträume ausgewählt wurde.



Den Betriebsstatus mit der Services-Tabelle überwachen

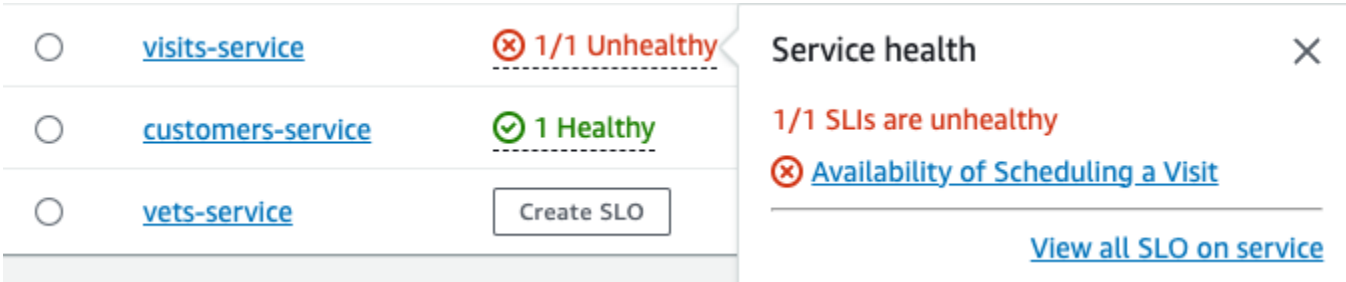
Auf der Services-Tabelle finden Sie eine Liste Ihrer Services, die für Application Signals aktiviert sind. Wählen Sie Application Signals aktivieren, um eine Einrichtungsseite zu öffnen und mit der Konfiguration Ihrer Services zu beginnen. Weitere Informationen finden Sie unter Application Signals aktivieren.

Filtern Sie die Services-Tabelle, um die Suche zu erleichtern, indem Sie eine oder mehrere Eigenschaften aus dem Filter-Textfeld auswählen. Bei der Auswahl der einzelnen Eigenschaften werden Sie durch die Filterkriterien geführt. Sie sehen den vollständigen Filter unter dem Filter-Textfeld. Sie können jederzeit Filter löschen auswählen, um den Tabellenfilter zu entfernen.



Wählen Sie den Namen eines beliebigen Services in der Tabelle aus, um eine Service-Detailseite mit Metriken, Vorgängen und zusätzlichen Details auf Service-Ebene anzuzeigen. Wenn Sie die dem Dienst zugrunde liegende Rechenressource einer Anwendung auf AppRegistry oder der Karte „Anwendungen“ auf der AWS Management Console Startseite zugeordnet haben, wählen Sie den Namen der Anwendung aus, um die Anwendungsdetails auf der Konsolenseite „Meine Anwendungen“ anzuzeigen. Wählen Sie für in Amazon EKS gehostete Services einen beliebigen Link in der Spalte Gehostet in, um Cluster, Namespace oder Workload in CloudWatch Container Insights anzuzeigen. Für Services, die in Amazon ECS oder Amazon EC2 ausgeführt werden, wird der Umgebungswert angezeigt.

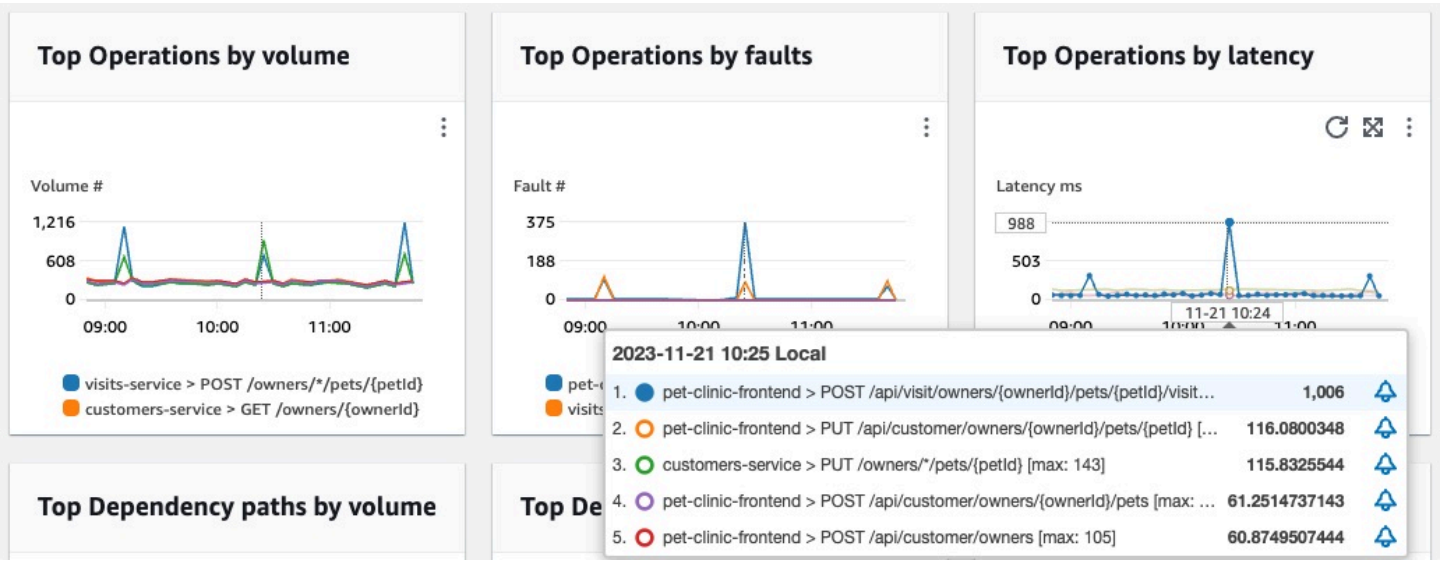
Der Status des Servicelevel-Indikators (SLI) wird für jeden Service in der Tabelle angezeigt. Wählen Sie den SLI-Status für einen Service, um ein Pop-up mit einem Link zu fehlerhaften SLIs und einem Link zu allen SLOs für den Service anzuzeigen.



Wenn für einen Service keine SLOs erstellt wurden, wählen Sie in der Spalte SLI-Status die Schaltfläche SLO erstellen. Um zusätzliche SLOs für einen Service zu erstellen, wählen Sie das Optionsfeld neben dem Namen des Services aus und wählen Sie dann oben rechts in der Tabelle SLO erstellen. Wenn Sie SLOs erstellen, können Sie auf einen Blick sehen, welche Ihrer Services und Vorgänge gut funktionieren und welche nicht. Weitere Informationen finden Sie unter Servicelevel-Ziele (SLOs).

Die wichtigsten Vorgangs- und Abhängigkeitsmetriken anzeigen

Unter der Services-Tabelle finden Sie die wichtigsten Vorgänge und Abhängigkeiten aller Services nach Aufrufvolumen, Störungen und Latenz. Diese Reihe von Diagrammen gibt Ihnen wichtige Informationen darüber, welche Vorgänge oder Abhängigkeiten bei allen Services möglicherweise fehlerhaft sind. Wählen Sie einen beliebigen Punkt in einem Diagramm, um ein Pop-up mit detaillierteren Serieninformationen zu öffnen. Bewegen Sie den Mauszeiger über die Serienbeschreibungen am unteren Rand eines Diagramms, um ein Pop-up mit detaillierten Metriken für einen bestimmten Vorgang oder einen bestimmten Abhängigkeitspfad anzuzeigen. Wählen Sie die Kontextmenü-Schaltfläche in der oberen rechten Ecke eines Diagramms, um weitere Optionen anzuzeigen, z. B. das Anzeigen von CloudWatch Metriken oder Protokollseiten.



Auf der Seite mit den Servicedetails können Sie detaillierte Serviceaktivitäten und den Betriebsstatus anzeigen

> ⚠ Application Signals befindet sich in der Vorschauversion für Amazon CloudWatch und kann sich ändern.

Wenn Sie Ihre Anwendung instrumentieren, ordnet Amazon CloudWatch Application Signals alle Dienste zu, die Ihre Anwendung erkennt. Verwenden Sie die Service-Detailseite, um einen Überblick über Ihre Services, Abläufe, Abhängigkeiten, Kanarien und Kundenanfragen für einen einzelnen Service zu erhalten. Gehen Sie wie folgt vor, um die Seite mit den Servicedetails aufzurufen:

- Öffnen Sie die CloudWatch -Konsole.
- Wählen Sie im linken Navigationsbereich im Abschnitt Anwendungssignale die Option Dienste aus.
- Wählen Sie den Namen eines beliebigen Dienstes aus den Tabellen „Dienste“, „Wichtigste Dienste“ oder „Abhängigkeiten“ aus.

Die Seite mit den Serviceinformationen ist in die folgenden Registerkarten unterteilt:

- Überblick — Verwenden Sie diese Registerkarte, um einen Überblick über einen einzelnen Service zu erhalten, einschließlich der Anzahl der Operationen, Abhängigkeiten, synthetischen Funktionen und Clientseiten. Auf der Registerkarte werden wichtige Kennzahlen für Ihren gesamten Service, die wichtigsten Operationen und Abhängigkeiten angezeigt. Zu diesen Metriken gehören Zeitreihendaten zu Latenz, Fehlern und Fehlern bei allen Servicevorgängen für diesen Service.
- Servicebetriebe — Verwenden Sie diese Registerkarte, um eine Liste der Vorgänge anzuzeigen, die Ihr Service aufruft, sowie interaktive Grafiken mit wichtigen Kennzahlen, die den Zustand der einzelnen Operationen messen. Sie können einen Datenpunkt in einem Diagramm auswählen, um Informationen zu Traces, Protokollen oder Metriken zu erhalten, die mit diesem Datenpunkt verknüpft sind.
- Abhängigkeiten — Verwenden Sie diese Registerkarte, um eine Liste der Abhängigkeiten, die Ihr Service aufruft, sowie eine Liste der Metriken für diese Abhängigkeiten anzuzeigen.
- Synthetics Canaries — Auf dieser Registerkarte finden Sie eine Liste von Synthetics Canaries, die Benutzeranrufe an Ihren Service simulieren, sowie wichtige Leistungskennzahlen für diese Canaries.
- Kundenseiten — Auf dieser Registerkarte finden Sie eine Liste der Kundenseiten, die Ihren Service aufrufen, sowie Messwerte, mit denen die Qualität der Kundeninteraktionen mit Ihrer Anwendung gemessen wird.

Sehen Sie sich Ihre Serviceübersicht an

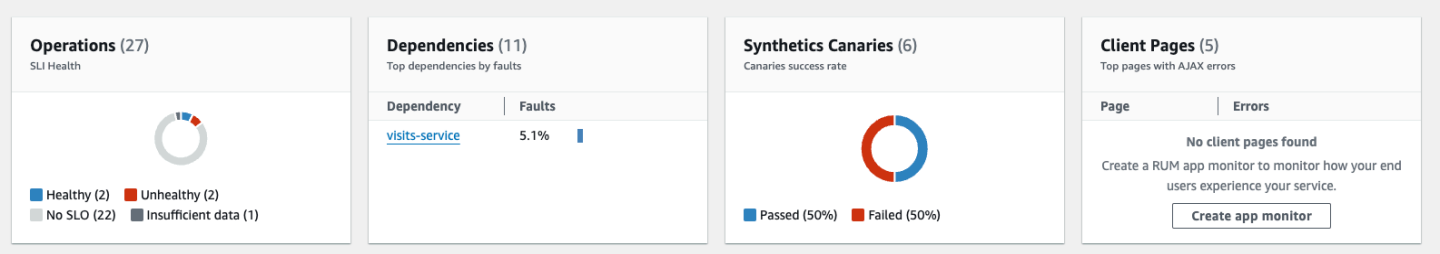
Verwenden Sie die Seite mit der Serviceübersicht, um eine allgemeine Zusammenfassung der Kennzahlen für alle Servicebetriebe an einem einzigen Standort anzuzeigen. Überprüfen Sie die Leistung aller Operationen, Abhängigkeiten, Client-Seiten und synthetischen Datenbanken, die mit Ihrer Anwendung interagieren. Anhand dieser Informationen können Sie ermitteln, worauf Sie sich konzentrieren sollten, um Probleme zu identifizieren, Fehler zu beheben und Optimierungsmöglichkeiten zu finden.

Wählen Sie einen beliebigen Link in den Servicedetails, um Informationen zu einem bestimmten Service anzuzeigen. Für Services, die in Amazon EKS gehostet werden, werden auf der Seite mit den Service-Details beispielsweise Cluster -, Namespace - und Workload-Informationen angezeigt. Für Services, die in Amazon ECS oder Amazon EC2 gehostet werden, wird auf der Seite mit den Servicedetails der Wert Environment angezeigt.

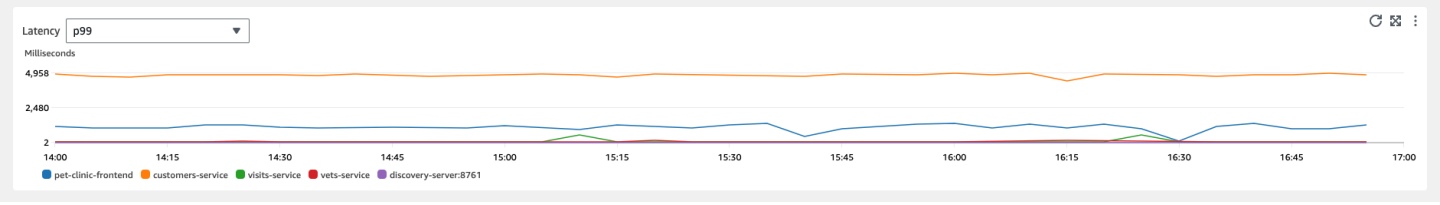
Unter Services wird auf der Registerkarte „Overview“ eine Zusammenfassung der folgenden Informationen angezeigt:

- Operationen — Verwenden Sie diese Registerkarte, um den Zustand Ihres Servicebetriebs zu überprüfen. Der Integritätsstatus wird durch Service Level Indicators (SLI) bestimmt, die als Teil eines Service Level Objective (SLO) definiert sind.
- Abhängigkeiten — In dieser Tabelle finden Sie die wichtigsten Abhängigkeiten der von Ihrer Anwendung aufgerufenen Dienste, sortiert nach Fehlerrate.
- Synthetics Canaries — Verwenden Sie diese Registerkarte, um das Ergebnis simulierter Aufrufe von Endpunkten oder APIs, die mit Ihrem Service verknüpft sind, sowie die Anzahl der fehlgeschlagenen Canaries anzuzeigen.
- Client-Seiten — Verwenden Sie diese Registerkarte, um die wichtigsten Seiten zu sehen, die von Clients aufgerufen wurden, bei denen asynchrone Fehler und XML- (AJAX JavaScript -) Fehler aufgetreten sind.

Die folgende Abbildung zeigt einen Überblick über Ihre Dienste:



Auf der Registerkarte „Übersicht“ wird auch ein Diagramm der Abhängigkeiten mit der höchsten Latenz für alle Dienste angezeigt. Verwenden Sie die Latenzmetriken p99, p90 und p50, um schnell zu beurteilen, welche Abhängigkeiten wie folgt zu Ihrer gesamten Servicelatenz beitragen:

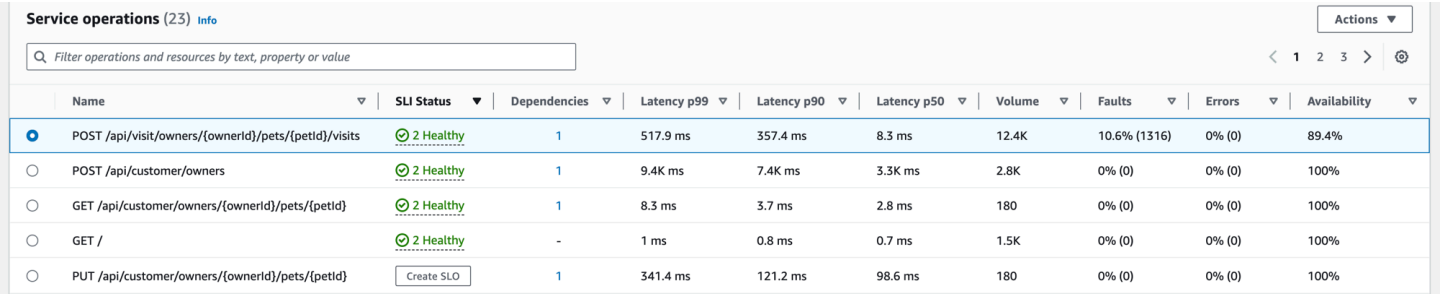


Das vorherige Diagramm zeigt beispielsweise, dass 99% der Anfragen, die an den Kundendienst gestellt wurden, in etwa 4.950 Millisekunden abgeschlossen wurden. Die anderen Abhängigkeiten nahmen weniger Zeit in Anspruch.

Diagramme, die die vier häufigsten Servicevorgänge nach Latenz darstellen, zeigen das Volumen der Anfragen, die Verfügbarkeit, die Fehlerrate und die Fehlerrate für diese Dienste, wie in der folgenden Abbildung dargestellt:

Ihre Service-Vorgänge anzeigen

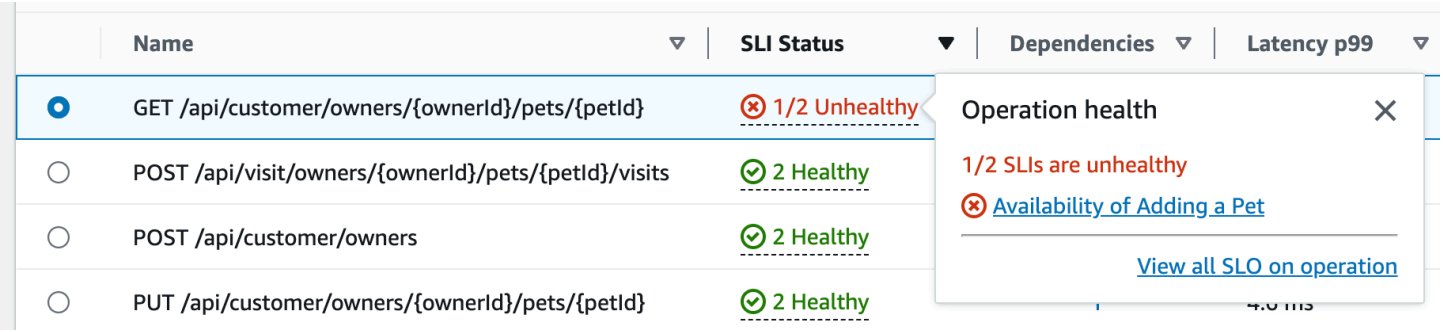
Wenn Sie Ihre Anwendung instrumentieren, erkennt Application Signals alle Serviceoperationen, die Ihre Anwendung aufruft. Verwenden Sie die Registerkarte Dienstvorgänge, um eine Tabelle mit den Dienstvorgängen und einer Reihe von Metriken anzuzeigen, mit denen die Leistung eines ausgewählten Vorgangs gemessen wird. Zu diesen Metriken gehören der SLI-Status, die Anzahl der Abhängigkeiten, die Latenz, das Volumen, Fehler, Fehler und Verfügbarkeit, wie in der folgenden Abbildung dargestellt:



Filtern Sie die Tabelle, um die Suche nach einem Servicevorgang zu erleichtern, indem Sie eine oder mehrere Eigenschaften aus dem Filtertextfeld auswählen. Bei der Auswahl der einzelnen Eigenschaften werden Sie durch die Filter-Kriterien geführt und der vollständige Filter wird unter dem

Filter-Textfeld angezeigt. Sie können jederzeit Filter löschen auswählen, um den Tabellenfilter zu entfernen.

Wählen Sie den SLI-Status für einen Vorgang, um ein Popup-Fenster mit einem Link zu einem fehlerhaften SLI und einem Link zu allen SLOs für den Vorgang anzuzeigen, wie in der folgenden Tabelle dargestellt:



In der Tabelle mit den Dienstvorgängen werden der SLI-Status, die Anzahl der fehlerfreien oder fehlerhaften SLIs und die Gesamtzahl der SLOs für jeden Vorgang aufgeführt.

Verwenden Sie SLIs, um Latenz, Verfügbarkeit und andere Betriebskennzahlen zu überwachen, die den Betriebsstatus eines Dienstes messen. Verwenden Sie ein SLO, um die Leistung und den Zustand Ihrer Dienste und Abläufe zu überprüfen.

Gehen Sie wie folgt vor, um ein SLO zu erstellen:

- Wenn ein Vorgang kein SLO hat, wählen Sie in der Spalte SLI-Status die Schaltfläche „SLO erstellen“.
- Wenn ein Vorgang bereits über einen SLO verfügt, gehen Sie wie folgt vor:
  - Wählen Sie das Optionsfeld neben dem Namen der Operation aus.
  - Wählen Sie mit dem Abwärtspfeil „Aktionen“ oben rechts in der Tabelle die Option „SLO erstellen“ aus.

Weitere Informationen finden Sie unter Servicelevel-Ziele (SLOs).

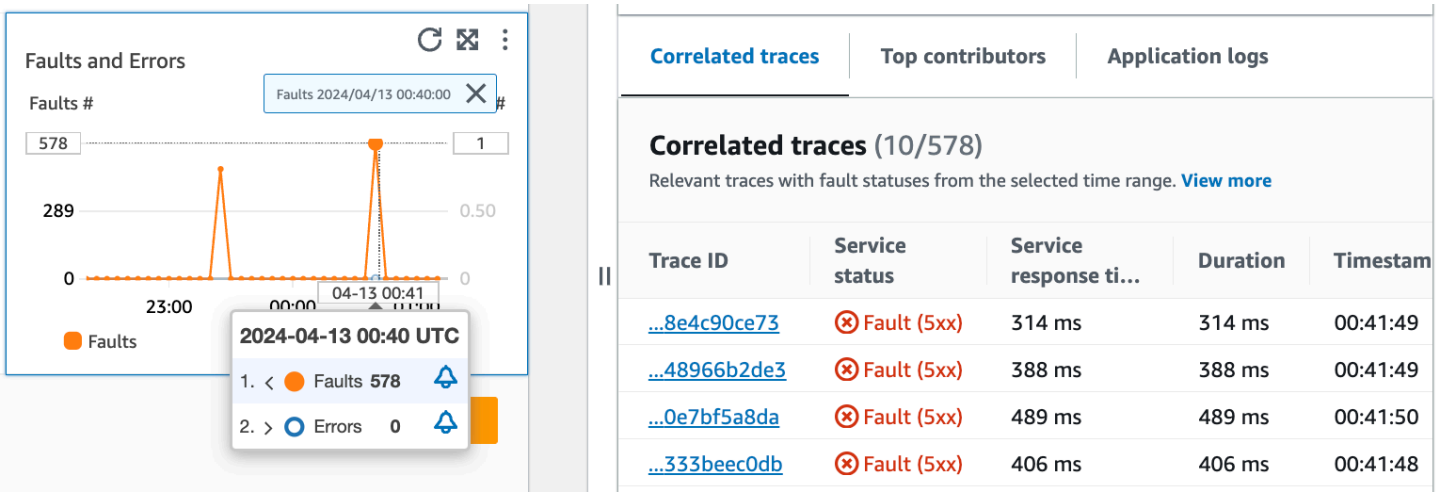
In der Spalte Abhängigkeiten wird die Anzahl der Abhängigkeiten angezeigt, die dieser Vorgang aufruft. Wählen Sie diese Zahl, um die Registerkarte Abhängigkeiten zu öffnen, die nach dem ausgewählten Vorgang gefiltert ist.

Zeigen Sie Kennzahlen für Servicebetriebe, korrelierte Ablaufverfolgungen und Anwendungsprotokolle an

Application Signals korreliert Servicebetriebsmetriken mit AWS X-Ray Traces, CloudWatch Container Insights und Anwendungsprotokollen. Verwenden Sie diese Metriken, um betriebliche Gesundheitsprobleme zu beheben. Gehen Sie wie folgt vor, um Metriken als grafische Informationen anzuzeigen:

1. Wählen Sie in der Tabelle Servicevorgänge einen Servicevorgang aus, um oberhalb der Tabelle eine Reihe von Diagrammen für den ausgewählten Vorgang mit Metriken für Volumen und Verfügbarkeit, Latenz sowie Fehler und Fehler anzuzeigen.
2. Zeigen Sie mit der Maus auf einen Punkt in einem Diagramm, um weitere Informationen anzuzeigen.
3. Wählen Sie einen Punkt aus, um einen Diagnosebereich zu öffnen, in dem korrelierte Traces, Metriken und Anwendungsprotokolle für den ausgewählten Punkt im Diagramm angezeigt werden.

Die folgende Abbildung zeigt den Tooltip, der angezeigt wird, wenn Sie den Mauszeiger über einen Punkt im Diagramm bewegen, und den Diagnosebereich, der angezeigt wird, wenn Sie auf einen Punkt klicken. Der Tooltip enthält Informationen über den zugehörigen Datenpunkt im Diagramm „Fehler und Fehler“. Der Bereich enthält korrelierte Ablaufverfolgungen, die wichtigsten Mitwirkenden und Anwendungsprotokolle, die dem ausgewählten Punkt zugeordnet sind.



Korrelierte Spuren

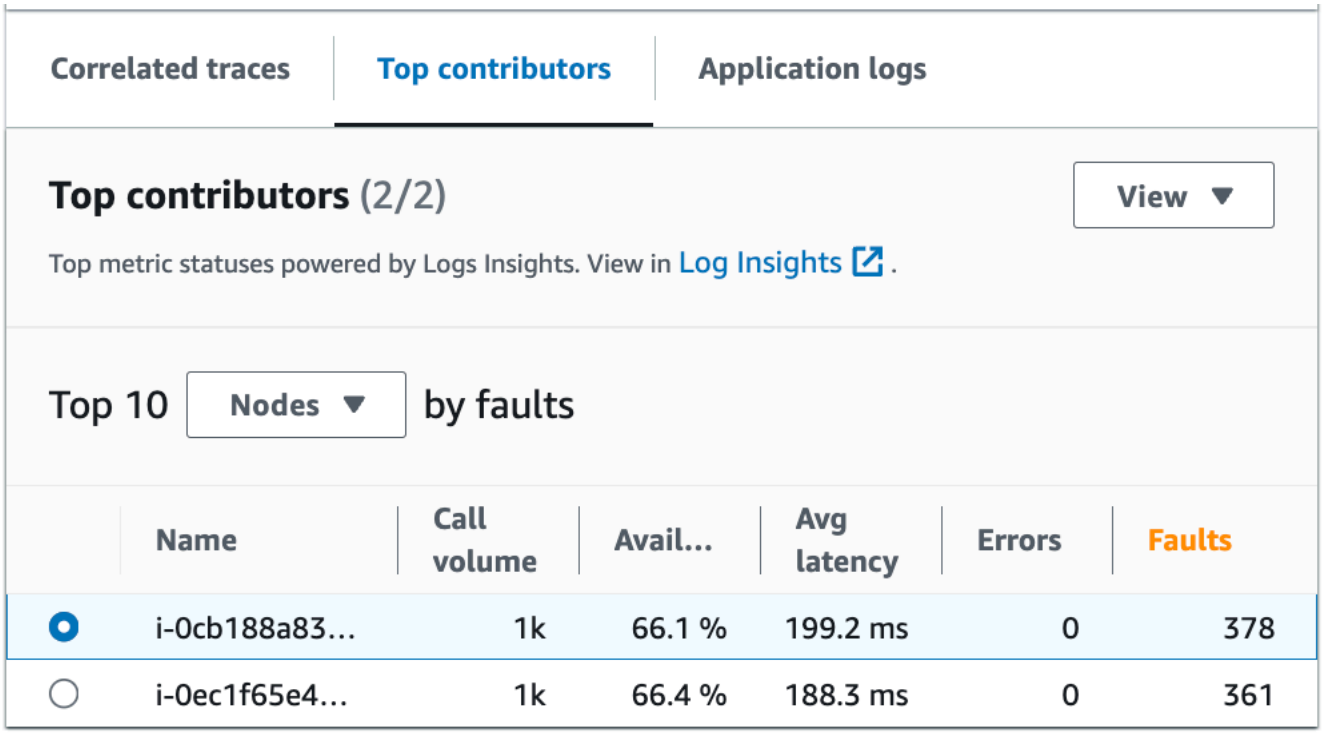
Schauen Sie sich verwandte Traces an, um zu verstehen, welches Problem einer Ablaufverfolgung zugrunde liegt. Sie können überprüfen, ob sich korrelierte Traces oder damit verknüpfte

Serviceknoten ähnlich verhalten. Um korrelierte Traces zu untersuchen, wählen Sie eine Trace-ID aus der Tabelle Korrelierte Traces aus, um die X-Ray-Trace-Detailseite für die gewählte Spur zu öffnen. Die Seite mit den Trace-Details enthält eine Übersicht der Service-Knoten, die der ausgewählten Trace zugeordnet sind, sowie eine Zeitleiste mit Trace-Segmenten.

Die wichtigsten Mitwirkenden

Sehen Sie sich die wichtigsten Mitwirkenden an, um die wichtigsten Eingabequellen für eine Metrik zu finden. Gruppieren Sie die Mitwirkenden nach verschiedenen Komponenten, um nach Ähnlichkeiten innerhalb der Gruppe zu suchen und zu verstehen, wie sich das Trace-Verhalten zwischen ihnen unterscheidet.

Auf der Registerkarte „Die häufigsten Mitwirkenden“ finden Sie Messwerte für Anrufvolumen, Verfügbarkeit, durchschnittliche Latenz, Fehler und Fehler für jede Gruppe. Das folgende Beispielbild zeigt die wichtigsten Beiträge zu einer Reihe von Metriken für eine Anwendung, die auf einer Amazon EKS-Plattform bereitgestellt wurde:



Die Liste der wichtigsten Mitwirkenden umfasst die folgenden Kennzahlen:

- Anrufvolumen — Verwenden Sie das Anrufvolumen, um die Anzahl der Anfragen pro Zeitintervall für eine Gruppe zu ermitteln.
- Verfügbarkeit — Verwenden Sie die Verfügbarkeit, um zu sehen, wie viel Prozent der Zeit für eine Gruppe keine Fehler festgestellt wurden.

- Durchschnittliche Latenz — Verwenden Sie die Latenz, um die durchschnittliche Dauer der Ausführung von Anfragen für eine Gruppe in einem Zeitintervall zu überprüfen, das davon abhängt, wie lange es her ist, dass die Anfragen, die Sie untersuchen, gestellt wurden. Anfragen, die vor weniger als 15 Tagen gestellt wurden, werden in Intervallen von 1 Minute ausgewertet. Anfragen, die zwischen 15 und einschließlich 30 Tagen gestellt wurden, werden in Intervallen von 5 Minuten bewertet. Wenn Sie beispielsweise Anfragen untersuchen, die vor 15 Tagen einen Fehler verursacht haben, entspricht die Kennzahl für das Anrufvolumen der Anzahl der Anfragen pro 5-Minuten-Intervall.
- Fehler — Die Anzahl der Fehler pro Gruppe, gemessen über ein Zeitintervall.
- Fehler — Die Anzahl der Fehler pro Gruppe über ein Zeitintervall.

Top-Beitragende, die Amazon EKS verwenden oder Kubernetes

Verwenden Sie Informationen über die wichtigsten Mitwirkenden für Anwendungen, die auf Amazon EKS bereitgestellt werdenKubernetes, oder um nach Knoten, Pod und Knoten gruppierte Betriebszustandsmetriken anzuzeigen PodTemplateHash. Es gelten die folgenden Definitionen:

- Ein Pod ist eine Gruppe von einem oder mehreren Docker Containern, die sich Speicher und Ressourcen teilen. Ein Pod ist die kleinste Einheit, die auf einer Kubernetes Plattform bereitgestellt werden kann. Gruppieren Sie nach Pods, um zu überprüfen, ob Fehler mit pod-spezifischen Einschränkungen zusammenhängen.
- Ein Knoten ist ein Server, auf dem Pods ausgeführt werden. Gruppieren Sie nach Knoten, um zu überprüfen, ob Fehler mit knotenspezifischen Einschränkungen zusammenhängen.
- Ein Pod-Template-Hash wird verwendet, um eine bestimmte Version einer Bereitstellung zu finden. Gruppieren Sie nach Pod-Vorlagen-Hash, um zu überprüfen, ob Fehler mit einer bestimmten Bereitstellung zusammenhängen.

Die besten Mitwirkenden, die Amazon EC2 verwenden

Verwenden Sie Informationen über die wichtigsten Mitwirkenden für Anwendungen, die auf Amazon EKS bereitgestellt werden, um Betriebszustandsmetriken, gruppiert nach Instance-ID und Auto Scaling-Gruppe, zu sehen. Es gelten die folgenden Definitionen:

- Eine Instance-ID ist eine eindeutige Kennung für die Amazon EC2 EC2-Instance, die Ihr Service ausführt. Gruppieren Sie nach Instance-ID, um zu überprüfen, ob Fehler mit einer bestimmten Amazon EC2 EC2-Instance zusammenhängen.

- Eine Auto Scaling-Gruppe ist eine Sammlung von Amazon EC2 EC2-Instances, mit denen Sie die Ressourcen, die Sie für die Bearbeitung Ihrer Anwendungsanfragen benötigen, nach oben oder unten skalieren können. Gruppieren Sie nach Auto Scaling-Gruppe, wenn Sie überprüfen möchten, ob Fehler auf die Instances innerhalb der Gruppe beschränkt sind.

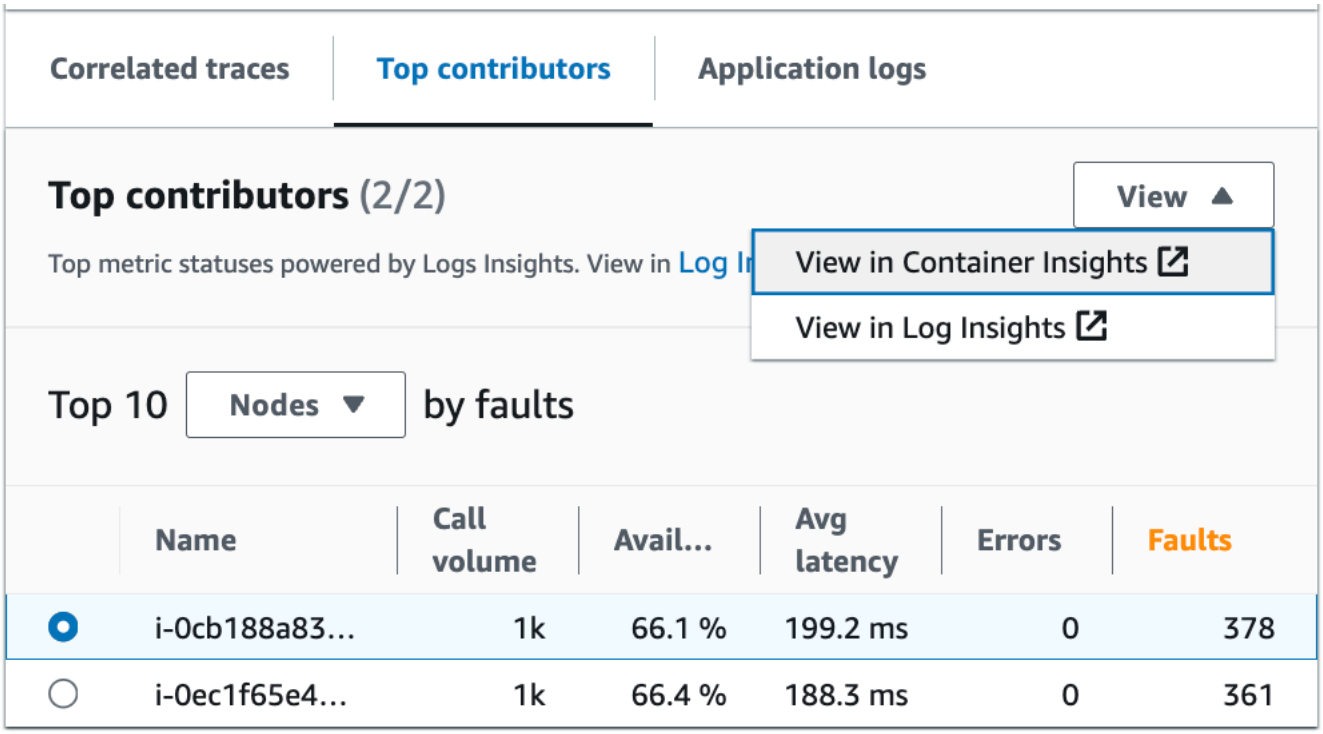
Die wichtigsten Mitwirkenden verwenden eine benutzerdefinierte Plattform

Verwenden Sie Informationen über die wichtigsten Mitwirkenden für Anwendungen, die mit benutzerdefinierter Instrumentierung bereitgestellt wurden, um nach Hostnamen gruppierte Kennzahlen zum Betriebsstatus anzuzeigen. Es gelten die folgenden Definitionen:

- Ein Hostname identifiziert ein Gerät wie einen Endpunkt oder eine Amazon EC2 EC2-Instance, die mit einem Netzwerk verbunden ist. Gruppieren Sie nach Hostnamen, um zu überprüfen, ob Ihre Fehler mit einem bestimmten physischen oder virtuellen Gerät zusammenhängen.

Sehen Sie sich die wichtigsten Mitwirkenden in Log Insights und an Container Insights

Die automatische Abfrage, die Kennzahlen für Ihre wichtigsten Mitwirkenden generiert hat, können Sie in Log Insights anzeigen und ändern. Sehen Sie sich in Container Insights Kennzahlen zur Infrastrukturleistung nach bestimmten Gruppen wie Pods oder Knoten an. Sie können Cluster, Knoten oder Workloads nach Ressourcenverbrauch sortieren und so Anomalien schnell identifizieren oder Risiken proaktiv mindern, bevor das Endbenutzererlebnis beeinträchtigt wird. Es folgt ein Bild, das zeigt, wie Sie diese Optionen auswählen können:



In Container Insights können Sie Metriken für Ihren Amazon EKS- oder Amazon ECS-Container anzeigen, die spezifisch für die Gruppierung Ihrer wichtigsten Mitwirkenden sind. Wenn Sie beispielsweise für einen EKS-Container nach Pod gruppiert haben, um die besten Mitwirkenden zu generieren, zeigt Container Insights Metriken und Statistiken an, die für Ihren Pod gefiltert wurden.

In Log Insights kannst du die Abfrage, die die Metriken generiert hat, unter Top-Beitragende wie folgt ändern:

1. Wählen Sie In Log Insights anzeigen aus. Die Logs Insights-Seite, die geöffnet wird, enthält eine Abfrage, die automatisch generiert wird und die folgenden Informationen enthält:

   - Der Name der Protokollcluster-Gruppe.
   - Der Vorgang, mit dem Sie die Untersuchung durchgeführt haben CloudWatch.
   - Das Aggregat der Metrik zur betrieblichen Integrität, mit der in der Grafik interagiert wurde.

   Die Protokollergebnisse werden automatisch gefiltert, sodass die Daten der letzten fünf Minuten angezeigt werden, bevor Sie den Datenpunkt im Service-Diagramm ausgewählt haben.
2. Um die Abfrage zu bearbeiten, ersetzen Sie den generierten Text durch Ihre Änderungen. Sie können den Abfragegenerator auch verwenden, um eine neue Abfrage zu generieren oder die vorhandene Abfrage zu aktualisieren.

Anwendungsprotokolle

Verwenden Sie die Abfrage auf der Registerkarte Anwendungsprotokolle, um protokollierte Informationen für Ihre aktuelle Protokollgruppe und Ihren aktuellen Dienst zu generieren und einen Zeitstempel einzufügen. Eine Protokollgruppe ist eine Gruppe von Protokollströmen, die Sie bei der Konfiguration Ihrer Anwendung definieren können.

Verwenden Sie eine Protokollgruppe, um Protokolle mit ähnlichen Merkmalen zu organisieren, darunter die folgenden:

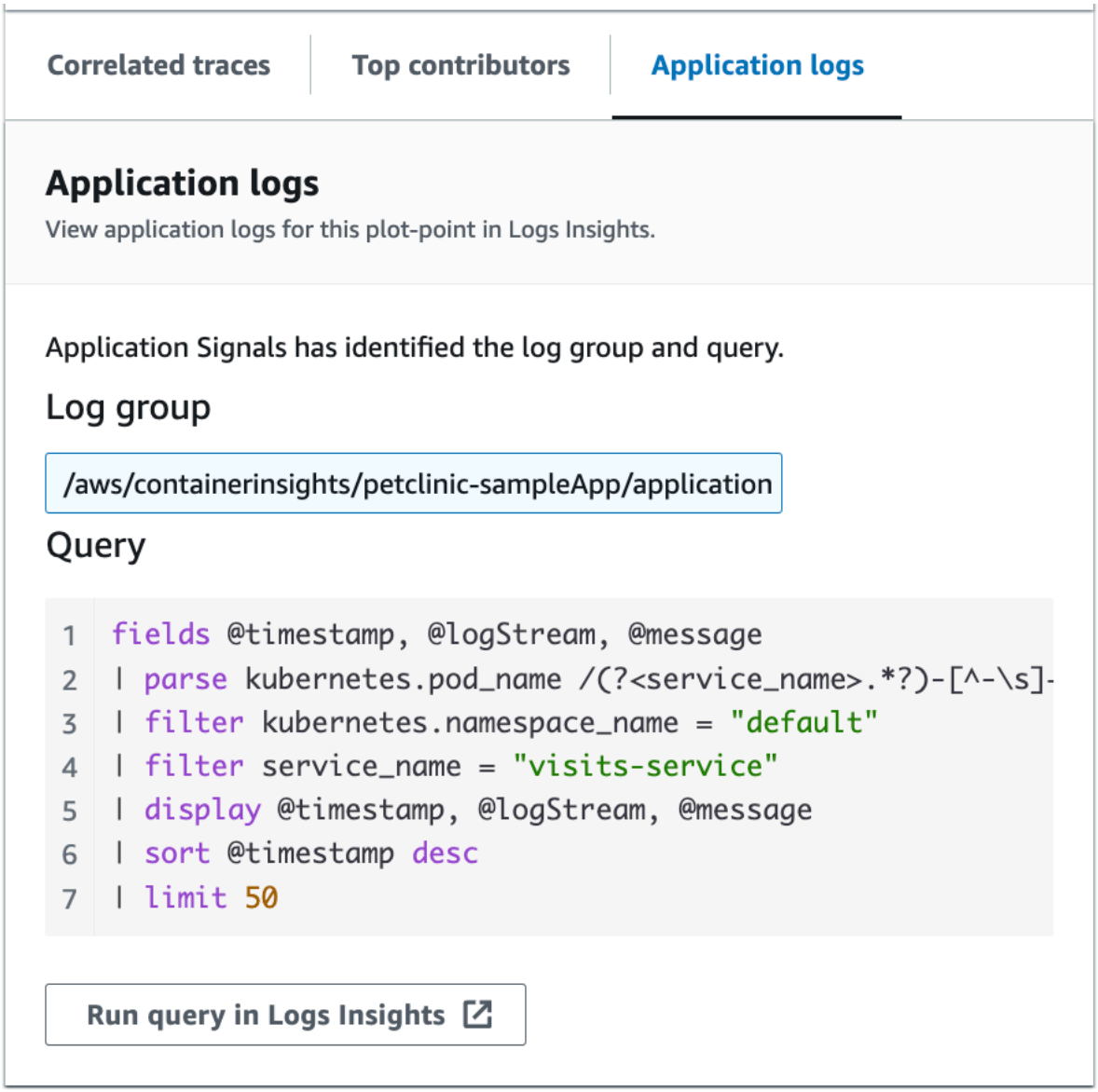
- Erfassen Sie Protokolle von einer bestimmten Organisation, Quelle oder Funktion.
- Erfassen Sie Protokolle, auf die ein bestimmter Benutzer zugreift.
- Erfassen Sie Protokolle für einen bestimmten Zeitraum.

Verwenden Sie diese Protokollstreams, um bestimmte Gruppen oder Zeitrahmen zu verfolgen. Sie können auch Überwachungsregeln, Alarme und Benachrichtigungen für diese Protokollgruppen einrichten. Weitere Informationen zu Protokollgruppen finden Sie unter Arbeiten mit Protokollgruppen und Protokollströmen.

Die Anwendungsprotokollabfrage gibt die Protokolle, wiederkehrende Textmuster und grafische Visualisierungen für Ihre Protokollgruppen zurück.

Um die Abfrage auszuführen, wählen Sie in Logs Insights die Option Abfrage ausführen aus, um entweder die automatisch generierte Abfrage auszuführen oder die Abfrage zu ändern. Um die Abfrage zu bearbeiten, ersetzen Sie den automatisch generierten Text durch Ihre Änderungen. Sie können den Abfragegenerator auch verwenden, um eine neue Abfrage zu generieren oder die vorhandene Abfrage zu aktualisieren.

Die folgende Abbildung zeigt die Beispielabfrage, die automatisch auf der Grundlage des ausgewählten Punkts im Servicebetriebsdiagramm generiert wird:



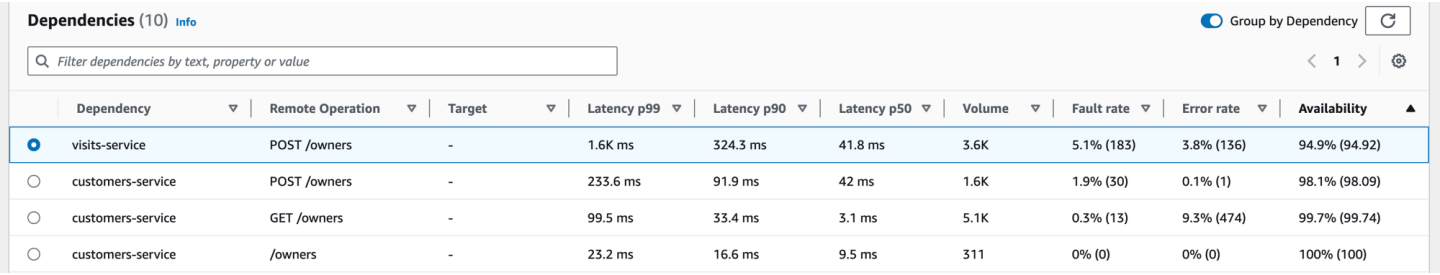
In der Abbildung oben CloudWatch hat die Protokollgruppe, die Ihrem ausgewählten Punkt zugeordnet ist, automatisch erkannt und in eine generierte Abfrage aufgenommen.

Ihre Service-Abhängigkeiten anzeigen

Wählen Sie die Registerkarte Abhängigkeiten, um die Abhängigkeiten-Tabelle und eine Reihe von Metriken für die Abhängigkeiten aller Service-Vorgänge oder eines einzelnen Vorgangs anzuzeigen. Die Tabelle enthält eine Liste der Abhängigkeiten, die von Application Signals erkannt wurden, einschließlich Metriken für Latenz, Aufrufvolumen, Störungsrate, Fehlerrate und Verfügbarkeit.

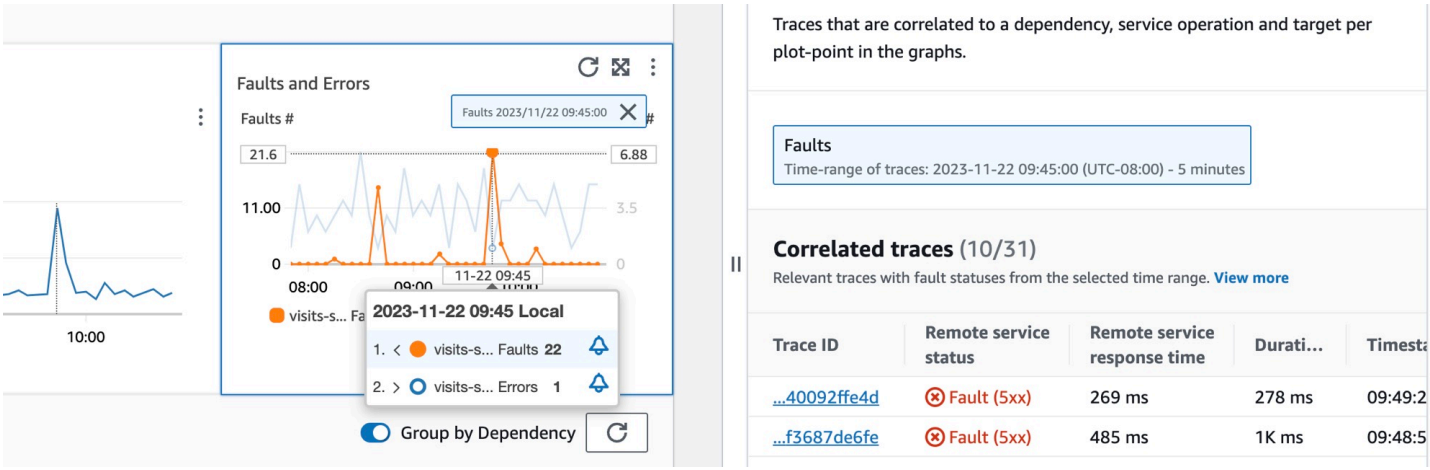
Wählen Sie oben auf der Seite einen Vorgang aus der Liste mit dem Abwärtspfeil aus, um die zugehörigen Abhängigkeiten anzuzeigen, oder wählen Sie Alle, um die Abhängigkeiten für alle Operationen anzuzeigen.

Filtern Sie die Tabelle, um die Suche nach dem, was Sie suchen, zu erleichtern, indem Sie eine oder mehrere Eigenschaften aus dem Filter-Textfeld auswählen. Bei der Auswahl der einzelnen Eigenschaften werden Sie durch die Filter-Kriterien geführt und der vollständige Filter wird unter dem Filter-Textfeld angezeigt. Sie können jederzeit Filter löschen auswählen, um den Tabellenfilter zu entfernen. Wählen Sie oben rechts in der Tabelle die Option Nach Abhängigkeit gruppieren aus, um Abhängigkeiten nach Service- und Vorgangsnamen zu gruppieren. Wenn die Gruppierung aktiviert ist, können Sie eine Gruppe von Abhängigkeiten mit dem +-Symbol neben dem Namen der Abhängigkeit erweitern oder reduzieren.



In der Spalte Abhängigkeit wird der Name des Abhängigkeits-Services angezeigt, während in der Spalte Remote-Vorgang der Name des Service-Vorgangs angezeigt wird. Beim Aufrufen von AWS Diensten wird in der Spalte Ziel die AWS Ressource angezeigt, z. B. die DynamoDB-Tabelle oder die Amazon SNS SNS-Warteschlange.

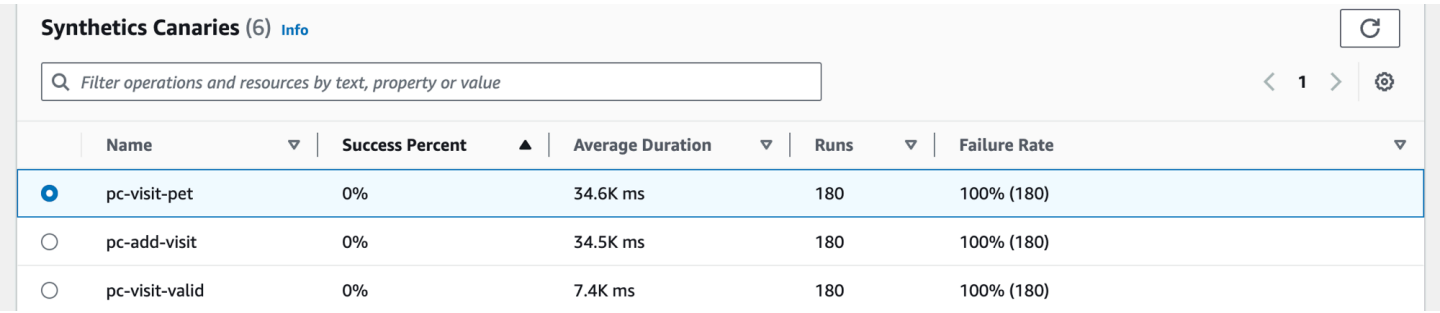
Um eine Abhängigkeit auszuwählen, wählen Sie in der Tabelle Abhängigkeiten die Option neben einer Abhängigkeit aus. Dies zeigt eine Reihe von Diagrammen, die detaillierte Metriken zu Anrufvolumen, Verfügbarkeit, Störungen und Fehlern anzeigen. Bewegen Sie den Mauszeiger über einen Punkt in einem Diagramm, um ein Popup mit weiteren Informationen zu sehen. Wählen Sie einen Punkt in einem Diagramm aus, um ein Diagnosefenster zu öffnen, in dem korrelierte Spuren für den ausgewählten Punkt im Diagramm angezeigt werden. Wählen Sie eine Trace-ID aus der Tabelle Korrelierte Traces aus, um die X-Ray-Trace-Detailseite für den ausgewählten Trace zu öffnen.



Ihre Synthetics-Canarys anzeigen

Wählen Sie die Registerkarte Synthetics-Canarys, um die Synthetics-Canarys-Tabelle und eine Reihe von Metriken für jeden Canary in der Tabelle anzuzeigen. Die Tabelle enthält Metriken für den prozentualen Erfolg, die durchschnittliche Dauer, die Ausführungen und die Ausfallrate. Es werden nur Kanarienvögel angezeigt, die für die AWS X-Ray Ablaufverfolgung aktiviert sind.

Verwenden Sie das Filter-Textfeld in der Tabelle mit synthetischen Kanarienvögeln, um den Kanarienvogel zu finden, der Sie interessiert. Jeder Filter, den Sie erstellen, wird unter dem Filtertextfeld angezeigt. Sie können jederzeit Filter löschen auswählen, um den Tabellenfilter zu entfernen.



Wählen Sie das Optionsfeld neben dem Namen des Kanarienvogels aus, um eine Reihe von Tabs mit Grafiken und detaillierten Kennzahlen wie Erfolgsquote, Fehlern und Dauer anzuzeigen. Bewegen Sie den Mauszeiger über einen Punkt in einem Diagramm, um ein Popup mit weiteren Informationen zu sehen. Wählen Sie einen Punkt in einem Diagramm aus, um ein Diagnosefenster zu öffnen, in dem Canary-Läufe angezeigt werden, die mit dem ausgewählten Punkt korrelieren. Wählen Sie einen Canary-Run und anschließend die Laufzeit aus, um Artefakte für den ausgewählten Canary-Run anzuzeigen, darunter Logs, HTTP Archivdateien (HAR), Screenshots und empfohlene Schritte

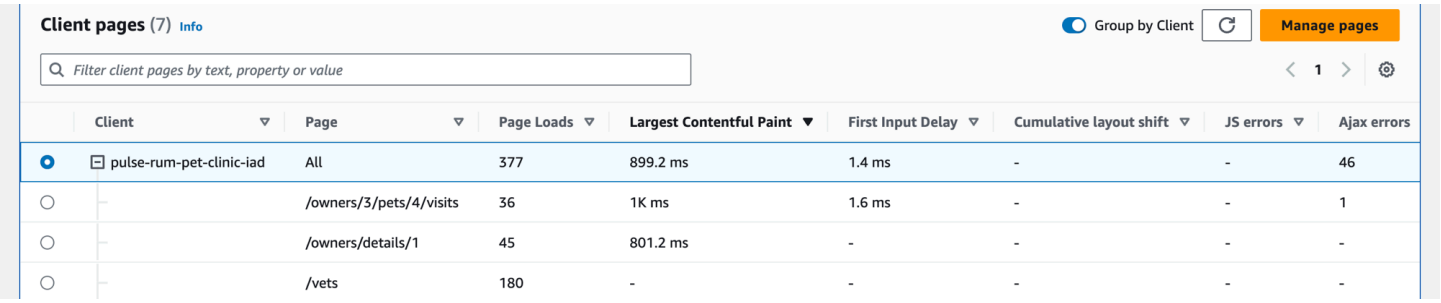
zur Problembehebung. Wähle Mehr erfahren, um die Seite CloudWatch Synthetics Canaries neben Canary Runs zu öffnen.

Ihre Kundenseiten anzeigen

Wählen Sie den Tab Client-Seiten, um eine Liste der Client-Webseiten anzuzeigen, die Ihren Service aufrufen. Verwenden Sie die Metriken für die ausgewählte Kundenseite, um die Qualität der Kundenerfahrung bei der Interaktion mit einem Service oder einer Anwendung zu messen. Zu diesen Metriken gehören Seitenladevorgänge, Web-Vitals und Fehler.

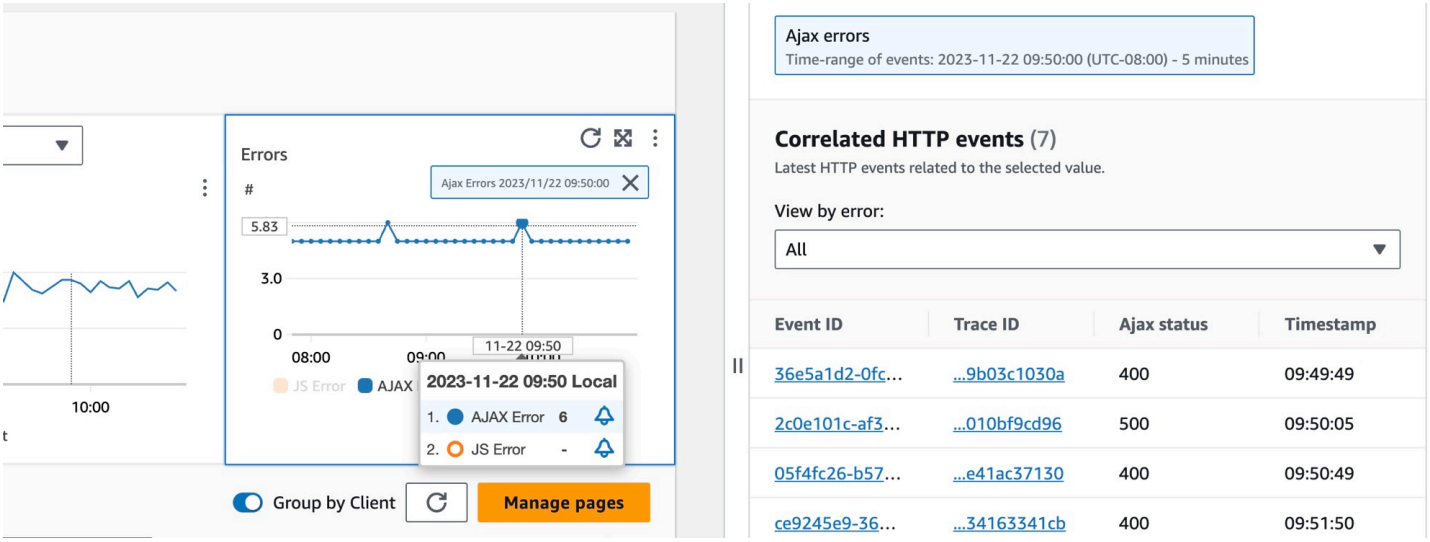
Um Ihre Kundenseiten in der Tabelle anzuzeigen, müssen Sie Ihren CloudWatch RUM-Webclient für X-Ray Tracing konfigurieren und Application Signals-Metriken für Ihre Kundenseiten aktivieren. Wählen Sie Seiten verwalten, um auszuwählen, welche Seiten für Application Signals-Metriken aktiviert sind.

Verwenden Sie das Filtertextfeld, um unter dem Filtertextfeld die Clientseite oder den Anwendungsmonitor zu finden, für den Sie sich interessieren. Wählen Sie Filter löschen, um den Tabellenfilter zu entfernen. Wählen Sie Nach Clients gruppieren, um Client-Seiten nach Clients zu gruppieren. Wenn Sie gruppiert sind, klicken Sie auf das +-Symbol neben einem Client-Namen, um die Zeile zu erweitern und alle Seiten für diesen Client anzuzeigen.



Um eine Client-Seite auszuwählen, wählen Sie in der Client-Seiten-Tabelle die Option neben einer Client-Seite aus. Sie werden eine Reihe von Diagrammen sehen, die detaillierte Metriken anzeigen. Bewegen Sie den Mauszeiger über einen Punkt in einem Diagramm, um ein Popup mit weiteren

Informationen zu sehen. Wählen Sie einen Punkt in einem Diagramm aus, um einen Diagnosebereich zu öffnen, in dem korrelierte Performance-Navigationsereignisse für den ausgewählten Punkt im Diagramm angezeigt werden. Wählen Sie eine Ereignis-ID aus der Liste der Navigationsereignisse aus, CloudWatch um die RUM-Seitenansicht für das gewählte Ereignis zu öffnen.



> **Note**
>
> Um AJAX-Fehler auf Ihren Kundenseiten zu sehen, verwenden Sie den CloudWatch RUM-Webclient Version 1.15 oder neuer.
> Derzeit können pro Service bis zu 100 Vorgangs-, Canary- und Client-Seiten sowie bis zu 250 Abhängigkeiten angezeigt werden.

Sehen Sie sich Ihre Anwendungstopologie an und überwachen Sie den Betriebsstatus mit der CloudWatch Service Map

> Application Signals befindet sich in der Vorschauversion für Amazon CloudWatch und kann sich ändern.

> **Note**
>
> Die CloudWatch Servicekarte ersetzt die ServiceLens Karte. Um eine auf AWS X-Ray Spuren basierende Karte Ihrer Anwendung zu sehen, öffnen Sie die X-Ray Trace Map. Wählen Sie

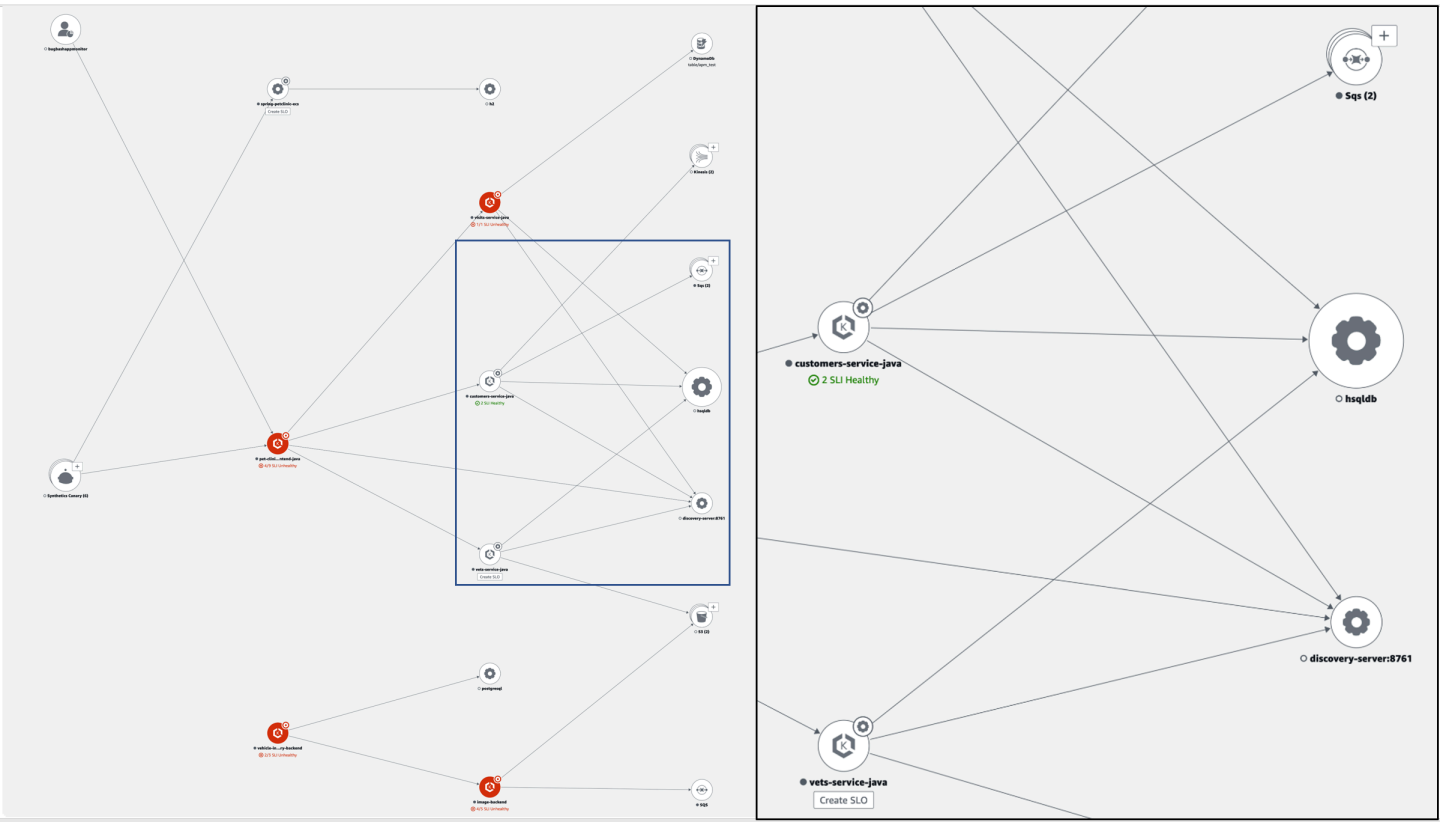
im linken Navigationsbereich der CloudWatch Konsole im Bereich X-Ray die Option Trace Map aus.

Verwenden Sie die Service-Map, um sich die Topologie Ihrer Anwendungsclients, Synthetics Canaries, Dienste und Abhängigkeiten anzusehen und den Betriebsstatus zu überwachen. Um die Service Map anzuzeigen, öffnen Sie die CloudWatch Konsole und wählen Sie im linken Navigationsbereich im Abschnitt Application Signals die Option Service Map aus.

Nachdem Sie Ihre Anwendung für Application Signals aktiviert haben, verwenden Sie die Service Map, um den Betriebsstatus Ihrer Anwendung einfacher zu überwachen:

- Sehen Sie sich Verbindungen zwischen Client-, Canary-, Service- und Abhängigkeitsknoten an, um Ihre Anwendungstopologie und den Ausführungsablauf besser zu verstehen. Dies ist besonders hilfreich, wenn Ihre Service-Anwender nicht Ihr Entwicklungsteam sind.
- Finden Sie heraus, welche Services Ihre Servicelevel-Ziele (SLOs) erfüllen oder nicht. Wenn ein Service Ihre SLOs nicht erfüllt, können Sie schnell erkennen, ob ein nachgelagerter Service oder eine Abhängigkeit möglicherweise zu dem Problem beiträgt oder sich auf mehrere Upstream-Services auswirkt.
- Wählen Sie einen einzelnen Client-, Synthetics Canary-, Service- oder Abhängigkeitsknoten aus, um die zugehörigen Metriken zu sehen. Auf der Seite mit den Servicedetails werden detailliertere Informationen zu Vorgängen, Abhängigkeiten, Synthetics Canaries und Client-Seiten angezeigt.
- Filtern und zoomen Sie die Service-Map, damit Sie sich leichter auf einen Teil Ihrer Anwendungstopologie konzentrieren oder die gesamte Map sehen können. Erstellen Sie einen Filter, indem Sie eine oder mehrere Eigenschaften aus dem Filter-Textfeld auswählen. Bei der Auswahl der einzelnen Eigenschaften werden Sie durch die Filterkriterien geführt. Sie sehen den vollständigen Filter unter dem Filter-Textfeld. Sie können jederzeit Filter löschen auswählen, um den Filter zu entfernen.

Die folgende Beispiel-Servicekarte zeigt Services mit Kanten, die sie mit Komponenten verbinden, mit denen sie interagieren. Wenn ein SLO definiert ist, zeigt die Service-Map auch den Integritätsstatus an.

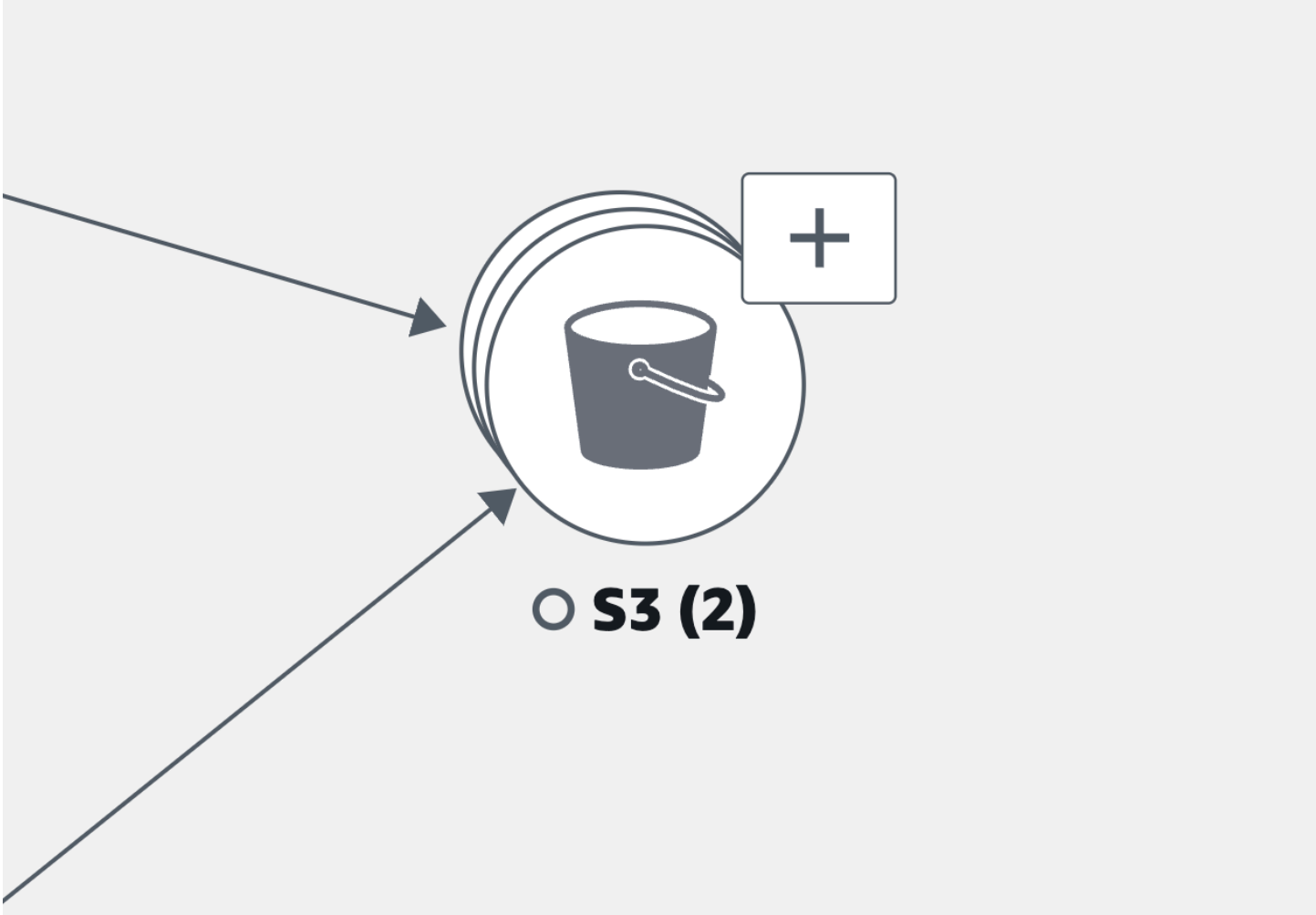


Erkunden Sie die Servicekarte

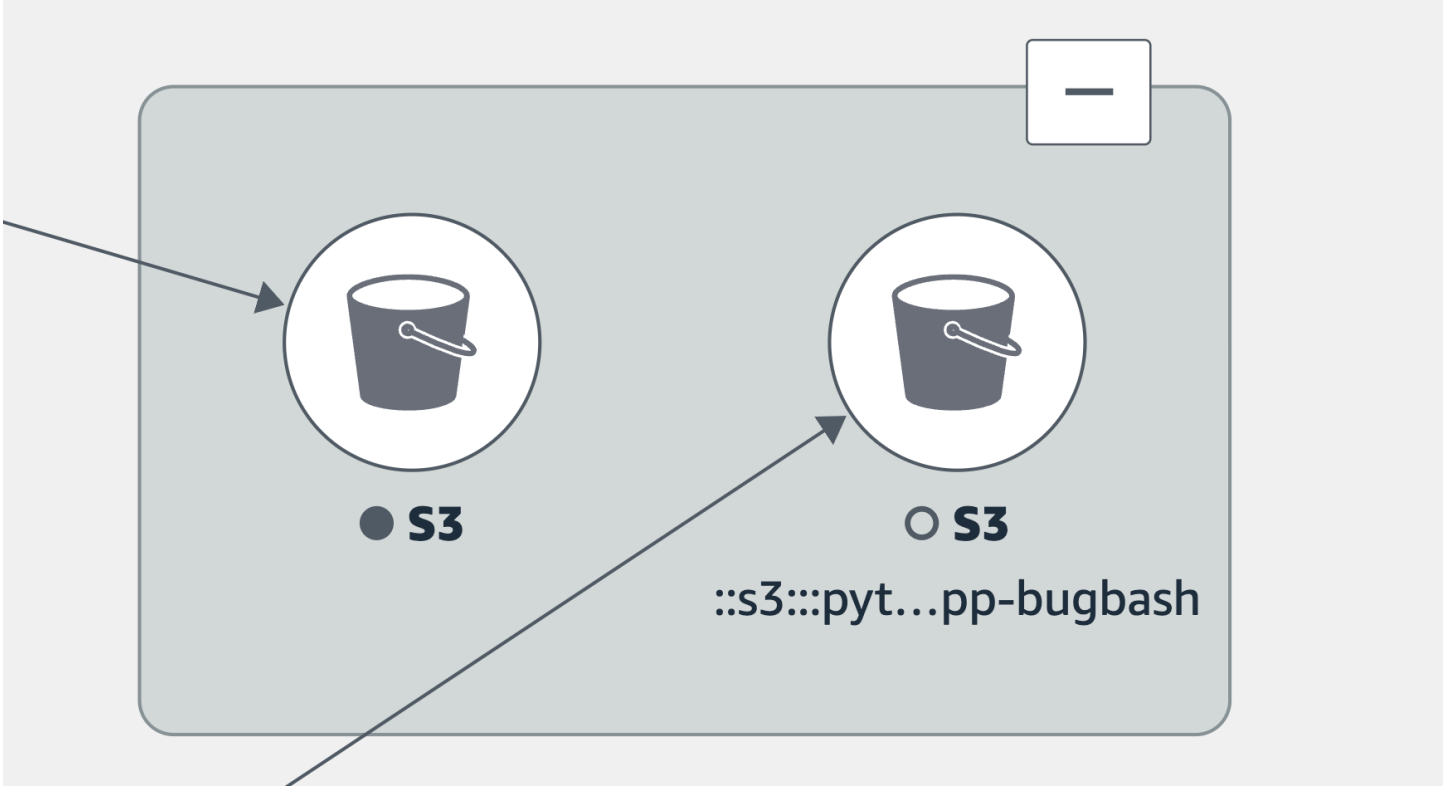
Nachdem Sie Ihre Anwendung für Application Signals aktiviert haben, zeigt die Service Map Knoten an, die Ihre Dienste und deren Abhängigkeiten repräsentieren.

Aktivieren Sie Active Tracing für Ihre CloudWatch RUM-Clients und Synthetics Canaries, um Client- und Canary-Nodes auf der Karte zu sehen.

Standardmäßig werden Canaries, RUM-Clients und AWS Serviceabhängigkeiten derselben Art in der Service Map zu einem einzigen erweiterbaren Symbol zusammengefasst. Dienstabhängigkeiten außerhalb von AWS werden standardmäßig nicht zusammen gruppiert. In der folgenden Abbildung sind beispielsweise alle Amazon S3 S3-Buckets unter einem erweiterbaren Symbol zusammengefasst:



In der vorherigen Abbildung zeigt die Bezeichnung zwischen der Amazon S3 S3-Gruppierung und dem ursprünglichen Service die Anzahl der Kanten zur Gruppe in Klammern unter dem Symbol der Abhängigkeit an. Wählen Sie das Symbol (+), um die Gruppe zu erweitern und ihre einzelnen Elemente zu sehen, wie in der folgenden Abbildung dargestellt:

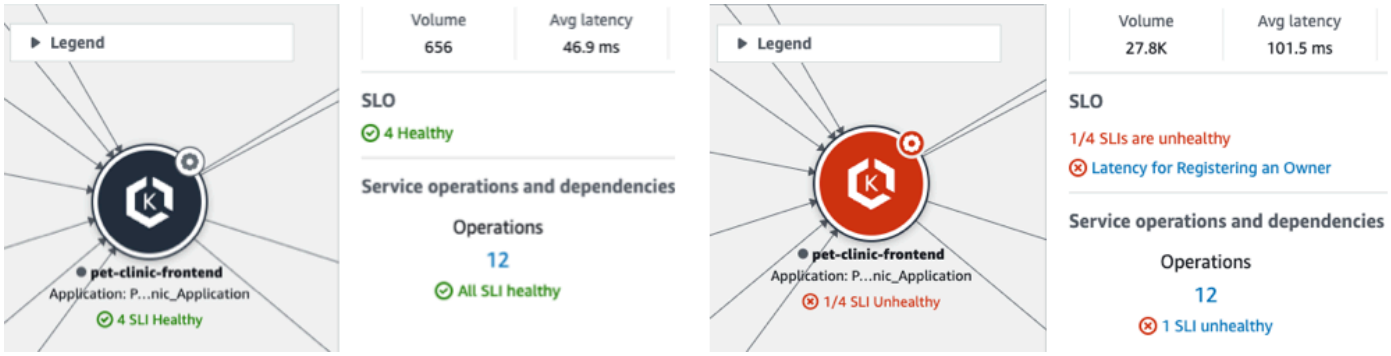


Wählen Sie eine Registerkarte, um Informationen zur Untersuchung der einzelnen Knotenarten und der Kanten (Verbindungen) zwischen ihnen zu erhalten.

View your application services

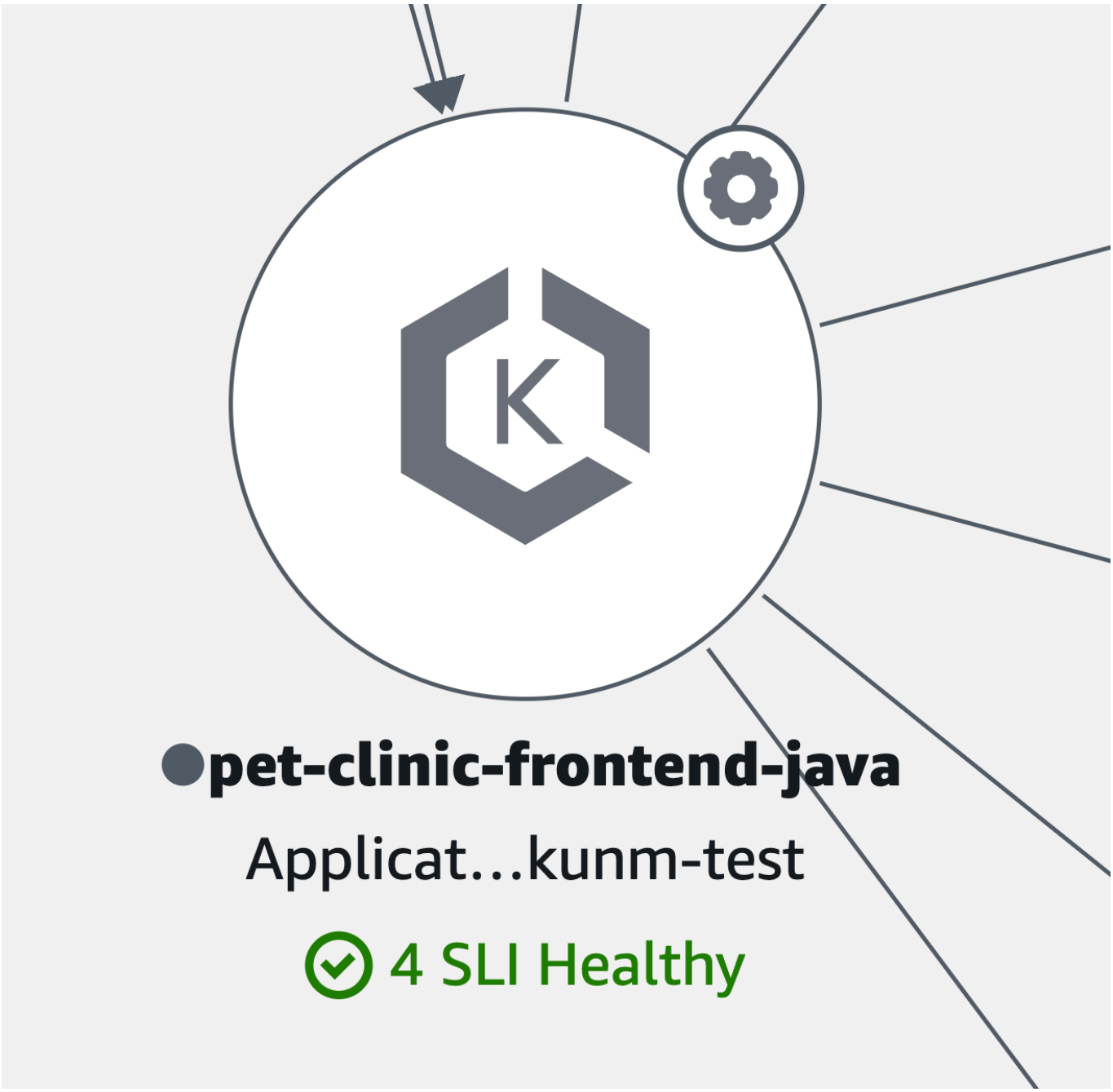
Sie können Ihre Anwendungsdienste und den Status ihrer SLOs und Service Level Indicators (SLIs) in der Service Map einsehen. Wenn Sie keine SLOs für einen Service erstellt haben, klicken Sie unter dem Serviceknoten auf die Schaltfläche „SLO erstellen“.

In der Service Map werden alle Ihre Dienste angezeigt. Außerdem werden die Kunden und Kanarienvögel angezeigt, die den Service nutzen, sowie die Abhängigkeiten, die von Ihren Diensten abhängig sind, wie in der folgenden Abbildung dargestellt:



Die folgenden Symbole stellen Beispiele für Anwendungsdienste in der Service Map dar:

- Amazon Elastic Kubernetes Service:



- Ein Kubernetes-Container:

- Amazon Elastic Compute Cloud (Amazon EC2):

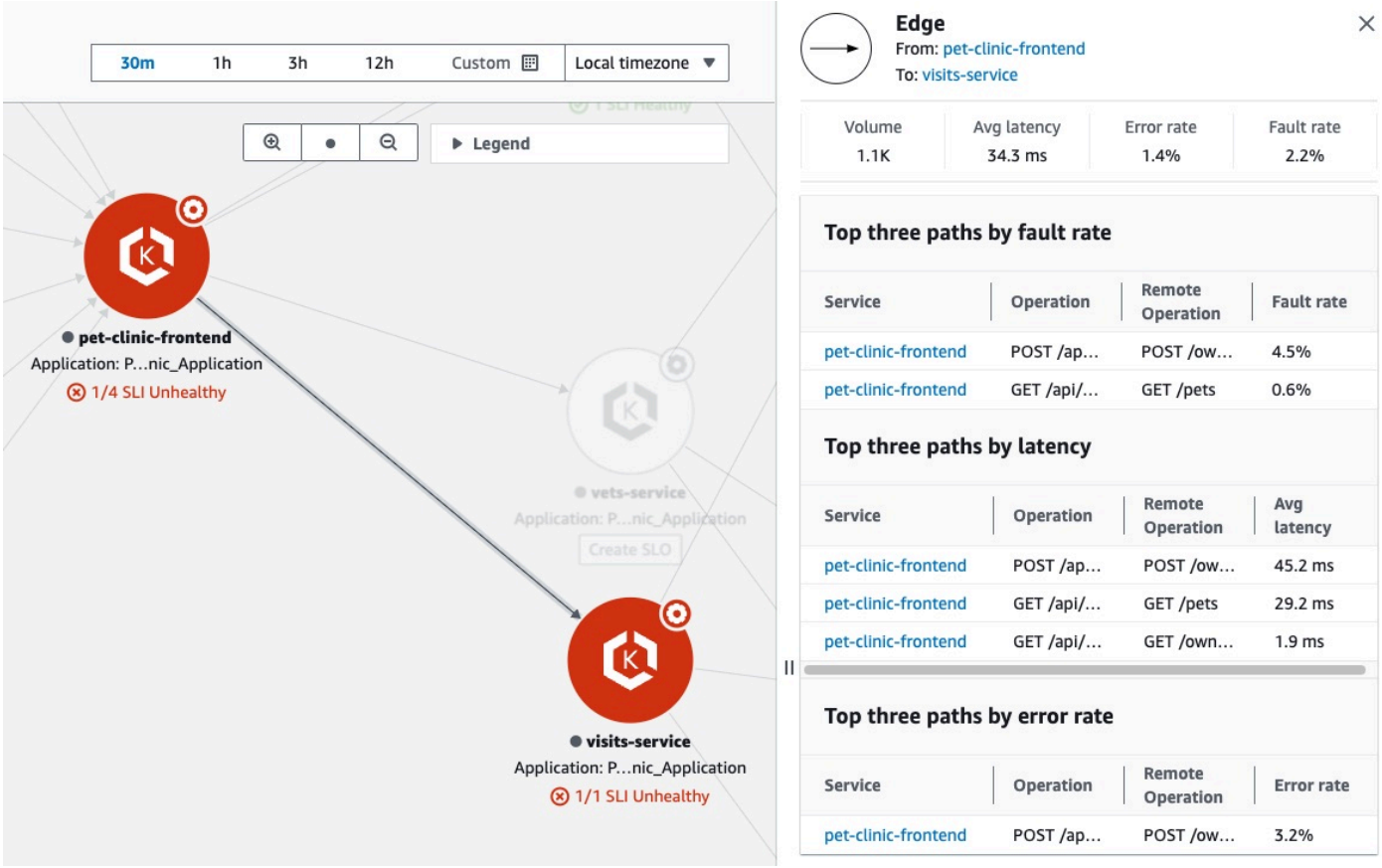
- Andere Anwendungsdiensttypen, die bisher nicht aufgeführt wurden:

Wenn Sie einen Dienstknoten auswählen, wird ein Bereich mit detaillierten Serviceinformationen geöffnet:

- Metriken für Anrufvolumen, Latenz, Fehler und Störungsraten.
- Die Anzahl der SLIs und SLOs, die oder sind`healthy`. `unhealthy`
- Die Option, weitere Informationen zu einem SLO anzuzeigen.
- Die Anzahl der Dienstoperationen, Abhängigkeiten, Synthetics Canaries und Client-Seiten.
- Die Option, jede Nummer auszuwählen, um die entsprechende Seite mit den Serviceinformationen zu öffnen.

- Der Anwendungsname, wenn Sie die zugrunde liegende Rechenressource einer Anwendung zugeordnet haben, die AppRegistry oder die Anwendungskarte auf der AWS Management Console Startseite verwendet.
  - Wählen Sie den Namen der Anwendung, um die Anwendungsdetails auf der myApplications-Konsolenseite anzuzeigen.
- Die`Cluster`, und `Workload` für Dienste`Namespace`, die in Amazon EKS gehostet werden, oder `Environment` für Dienste, die in Amazon ECS oder Amazon EC2 gehostet werden. Wählen Sie für von Amazon EKS gehostete Dienste einen beliebigen Link, um CloudWatch Container Insights zu öffnen.

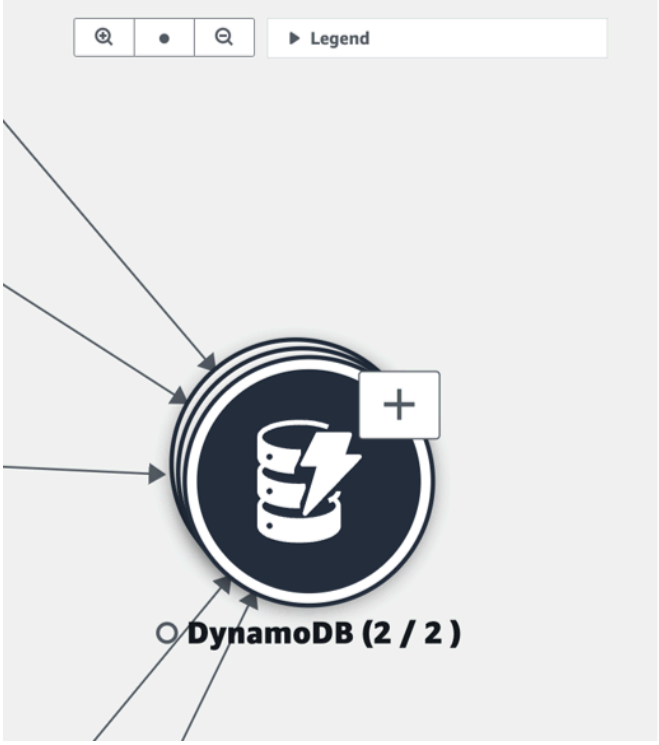
Wählen Sie einen Edge oder eine Verbindung zwischen einem Serviceknoten und einem Downstream-Service- oder Abhängigkeitsknoten aus. Dadurch wird ein Bereich geöffnet, der die wichtigsten Pfade nach Fehlerrate, Latenz und Fehlerrate enthält, wie im folgenden Beispielbild dargestellt. Wählen Sie einen beliebigen Link im Bereich, um die Seite mit den Servicedetails zu öffnen und detaillierte Informationen zum ausgewählten Dienst oder zur ausgewählten Abhängigkeit anzuzeigen.



View dependencies

Ihre Anwendungsabhängigkeiten werden auf der Service Map angezeigt und sind mit den Diensten verbunden, die sie aufrufen.

Wählen Sie einen Abhängigkeitsknoten aus, um einen Bereich zu öffnen, der die wichtigsten Pfade nach Fehlerrate, Latenz und Fehlerrate enthält. Wählen Sie einen beliebigen Service- oder Ziel-Link aus, um die Seite mit den Servicedetails zu öffnen und detaillierte Informationen zum ausgewählten Service- oder Abhängigkeitsziel anzuzeigen, wie in der folgenden Beispielabbildung dargestellt:



Volume	Avg latency	Error rate	Fault rate
-	-	-	-

Top three paths by fault rate

Service	Remote operation	Fault rate
No paths with faults		

Top three paths by latency

Service	Remote operation	Avg latency
billing-service-ec2-python	PutItem	282.8 ms
billing-service-python	PutItem	75.6 ms
visits-service-java	PutItem	64.9 ms

Top three paths by error rate

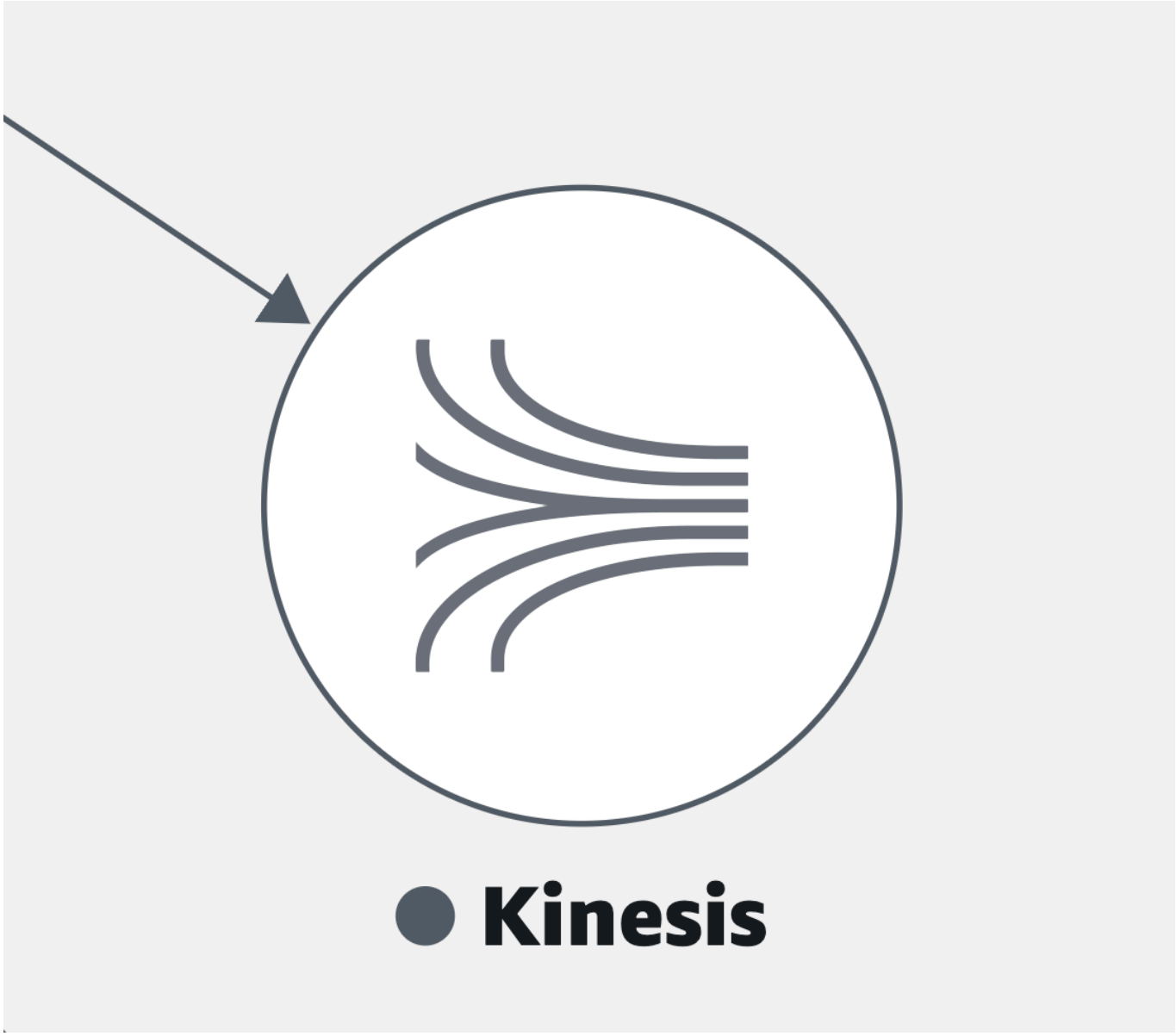
Service	Remote operation	Error rate
visits-service-java	PutItem	9.6%

Dienstabhängigkeiten werden standardmäßig in einem einzigen erweiterbaren Symbol zusammengefasst. Wählen Sie das Symbol (+), wie in der vorherigen Abbildung gezeigt, um die Gruppe zu erweitern und ihre einzelnen Elemente zu sehen.

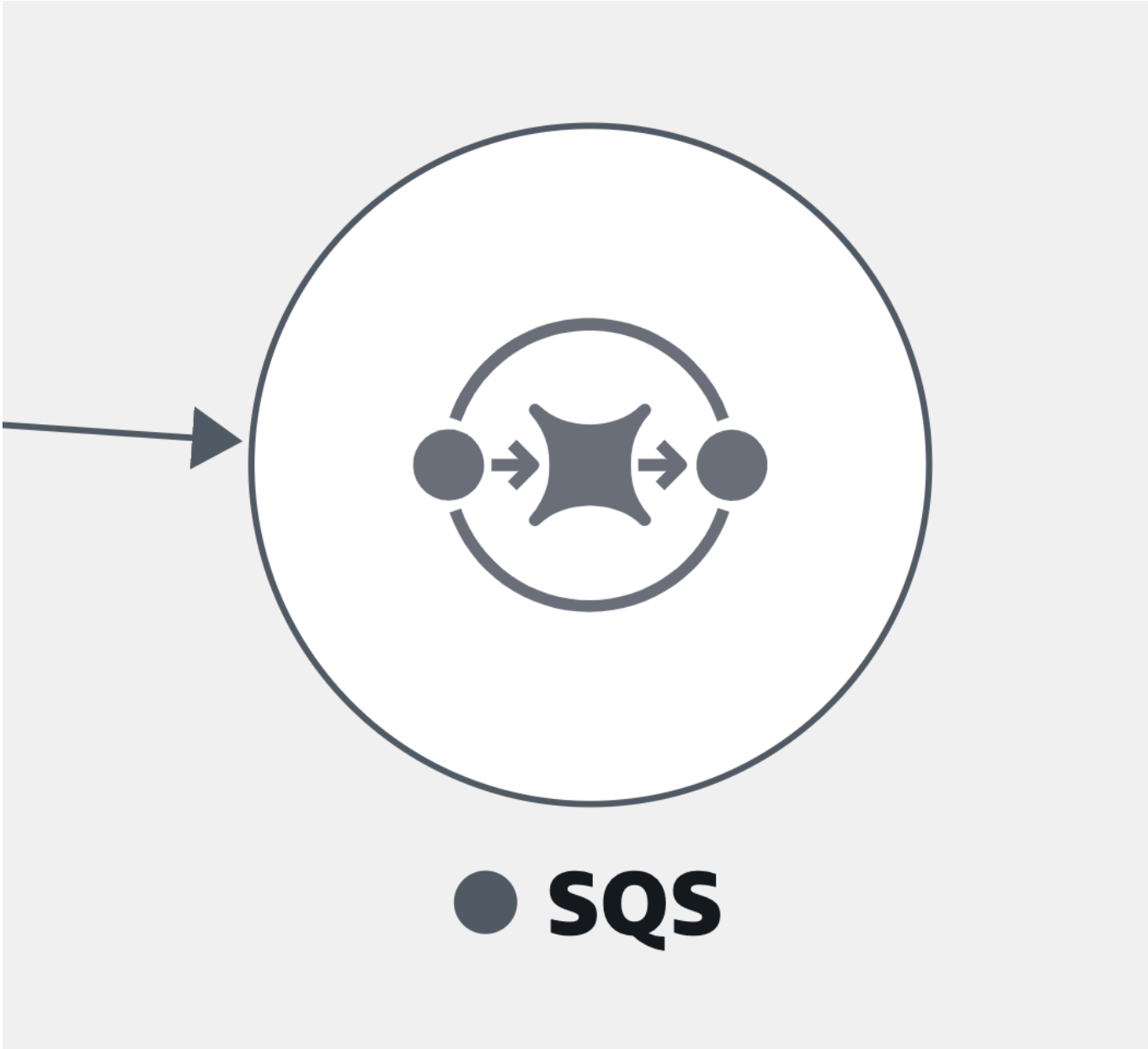
Die folgenden Symbole stellen Beispiele für Abhängigkeitsknoten in der Service-Map dar:

- Ein Amazon S3 S3-Bucket:

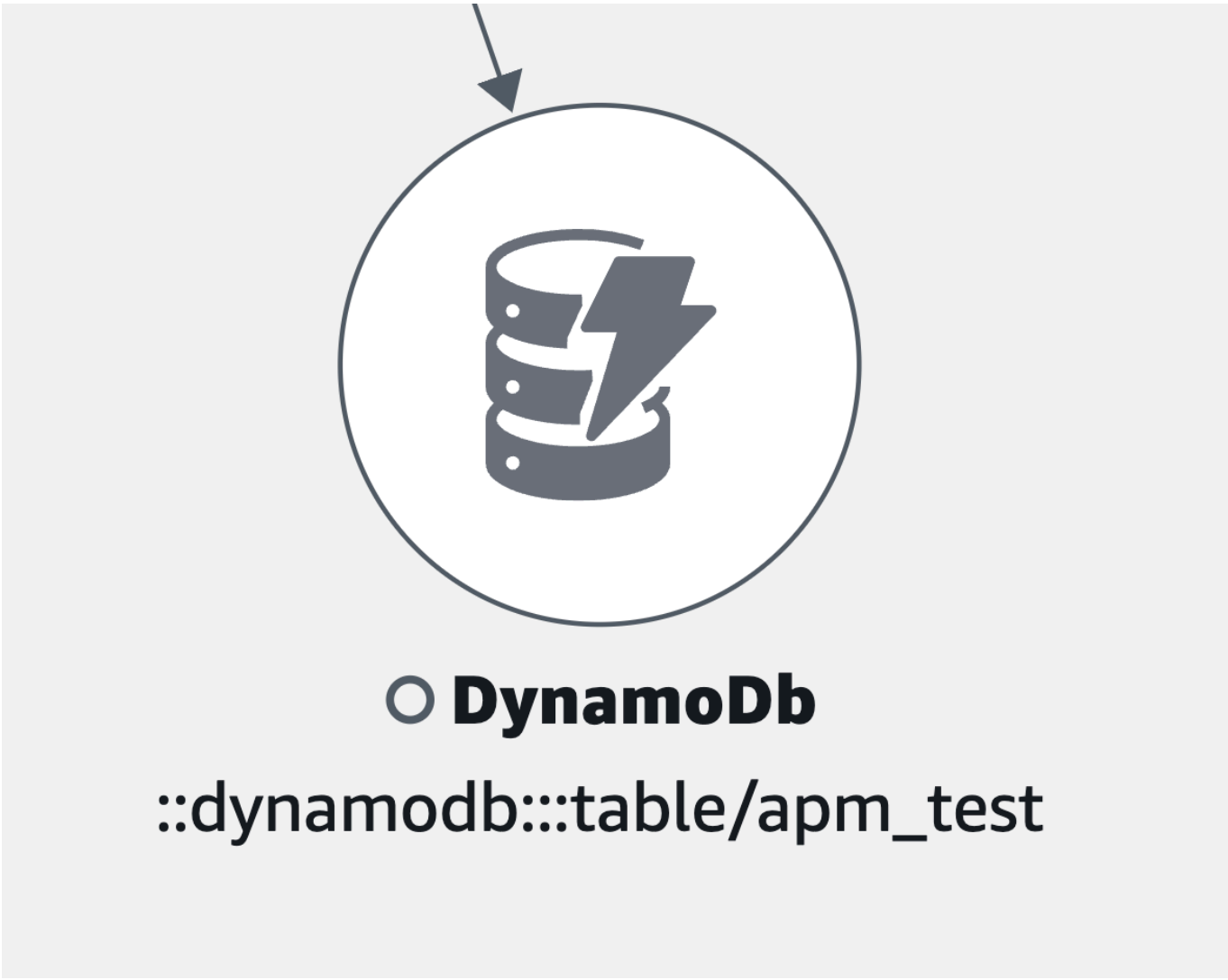
- Ein Amazon Kinesis Kinesis-Stream:



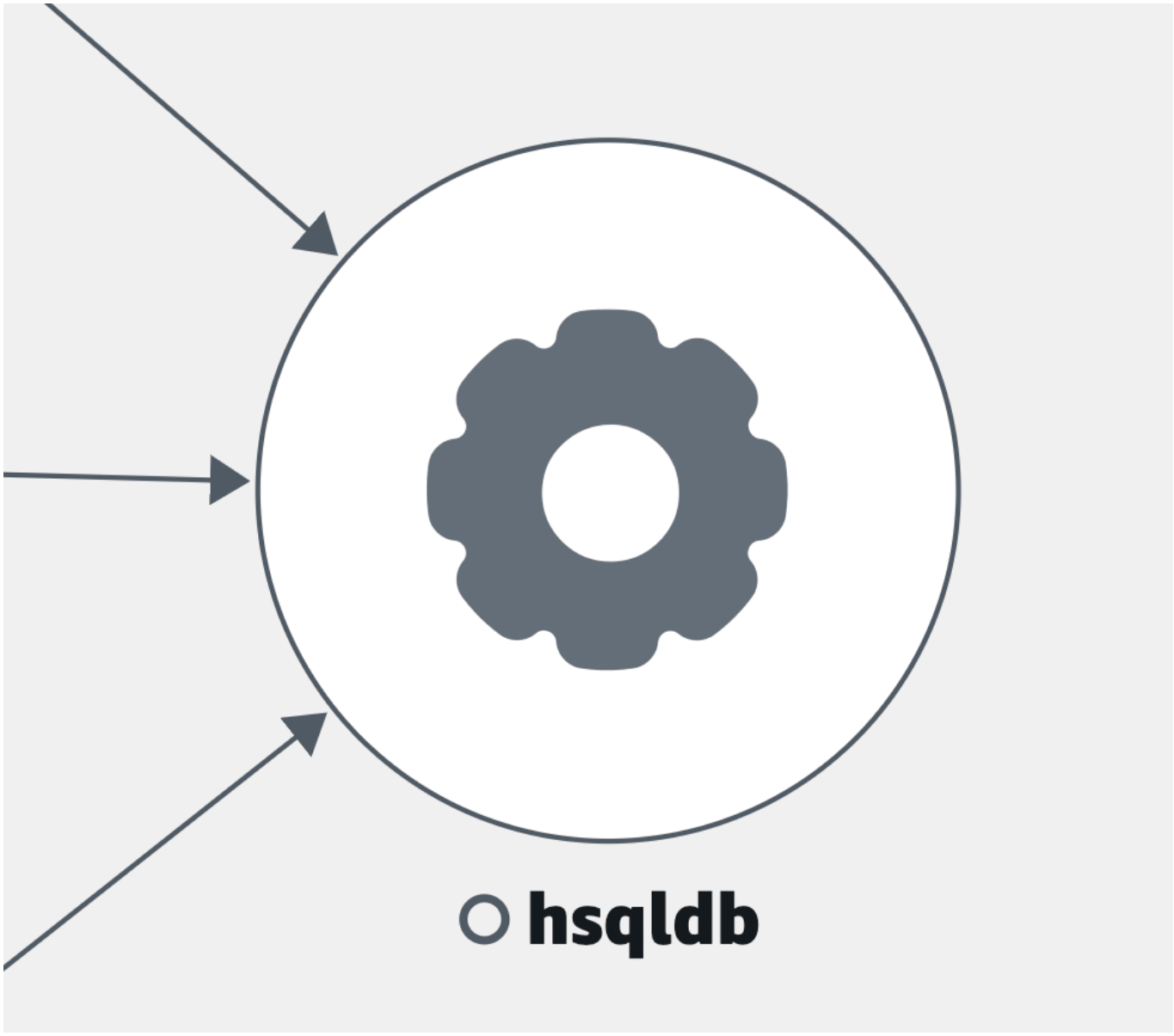
- Amazon Simple Queue Service (Amazon SQS):



- Eine Amazon DynamoDB-Tabelle:



- Andere Abhängigkeitstypen, die bisher nicht aufgeführt wurden:



View clients

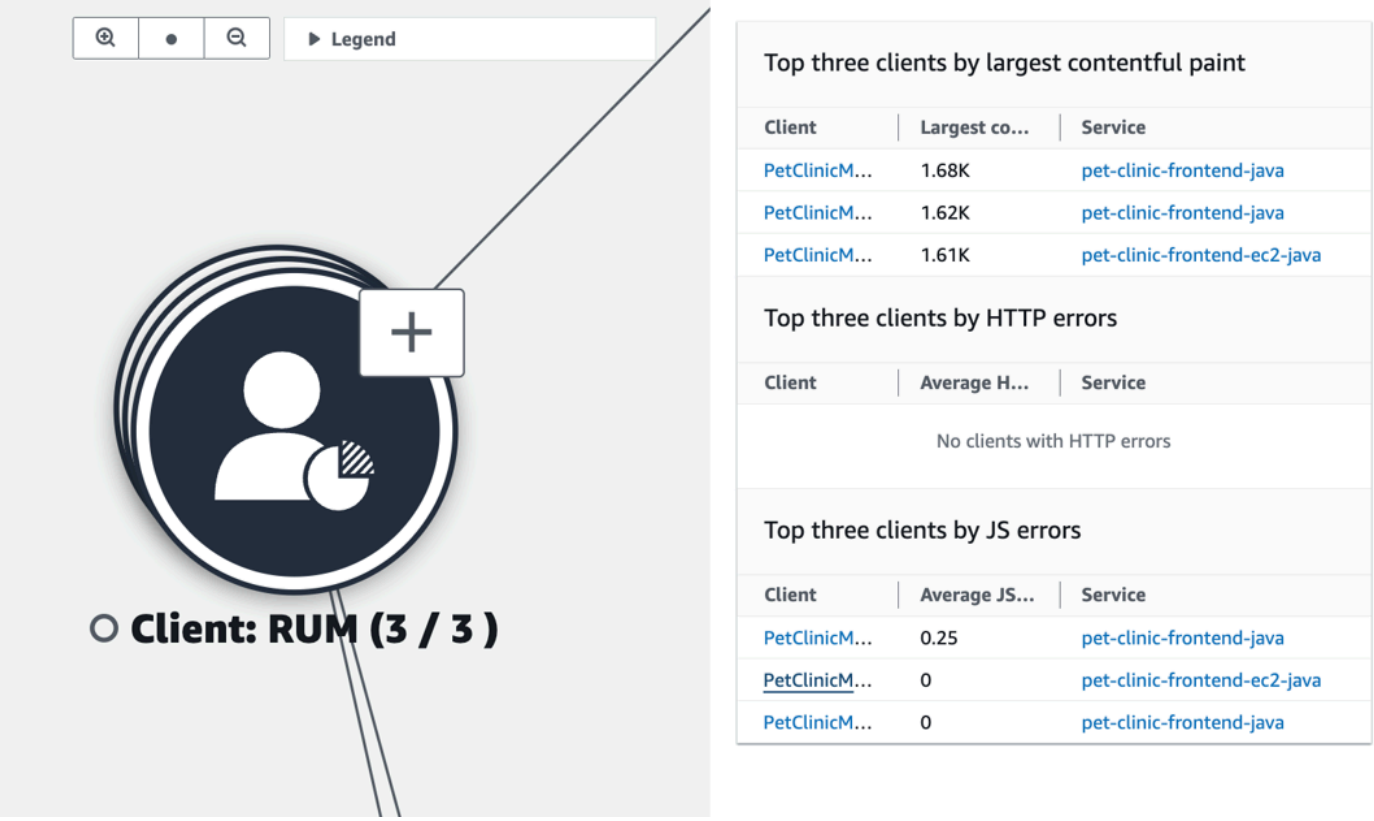
Nachdem Sie X-Ray Tracing für Ihre CloudWatch RUM-Webclients aktiviert haben, werden sie auf der Service-Map angezeigt, die mit den von ihnen aufgerufenen Diensten verbunden sind.

Wählen Sie einen Client-Knoten aus, um einen Bereich mit detaillierten Client-Informationen zu öffnen:

- Metriken für Seitenladevorgänge, durchschnittliche Ladezeit, Fehler und durchschnittliche Webdaten.
- Ein Diagramm, das eine Aufschlüsselung der Fehler anzeigt.

- Ein Link zur Anzeige der Kundendetails in CloudWatch RUM.

RUM-Clients sind standardmäßig zu einem einzigen erweiterbaren Symbol zusammengefasst. Wählen Sie das Symbol (+), wie in der folgenden Abbildung gezeigt, um die Gruppe zu erweitern und ihre einzelnen Elemente zu sehen.



Das folgende Symbol stellt ein Beispiel für einen RUM-Client in der Service Map dar:

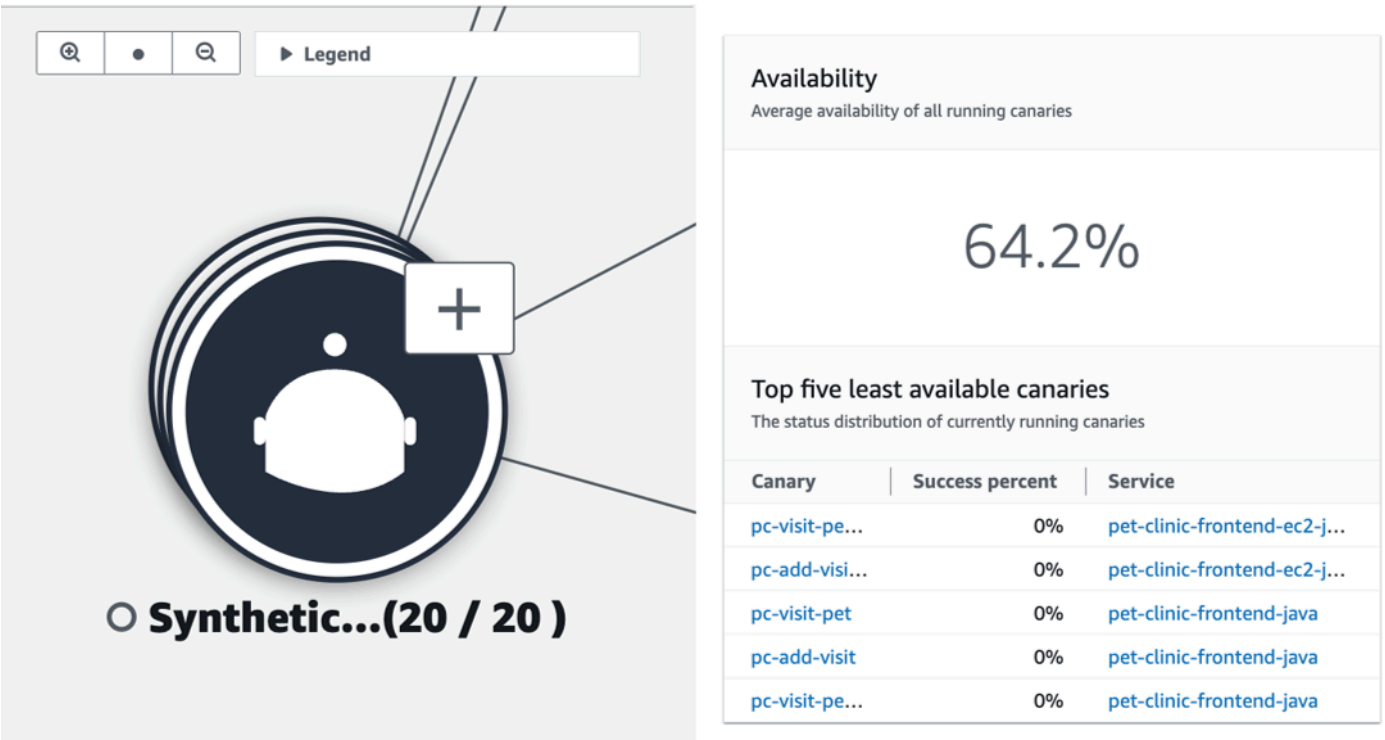
- Ein RUM-Client —

> **Note**
>
> Um AJAX-Fehler auf Ihren Kundenseiten zu sehen, verwenden Sie den CloudWatch RUM-Webclient Version 1.15 oder neuer.

View synthetics canaries

Nachdem Sie die AWS X-Ray Ablaufverfolgung für Ihre CloudWatch Synthetics Canaries aktiviert haben, werden sie auf der Service-Map angezeigt, die mit den von ihnen aufgerufenen Diensten verbunden ist, wie im folgenden Beispielbild dargestellt:

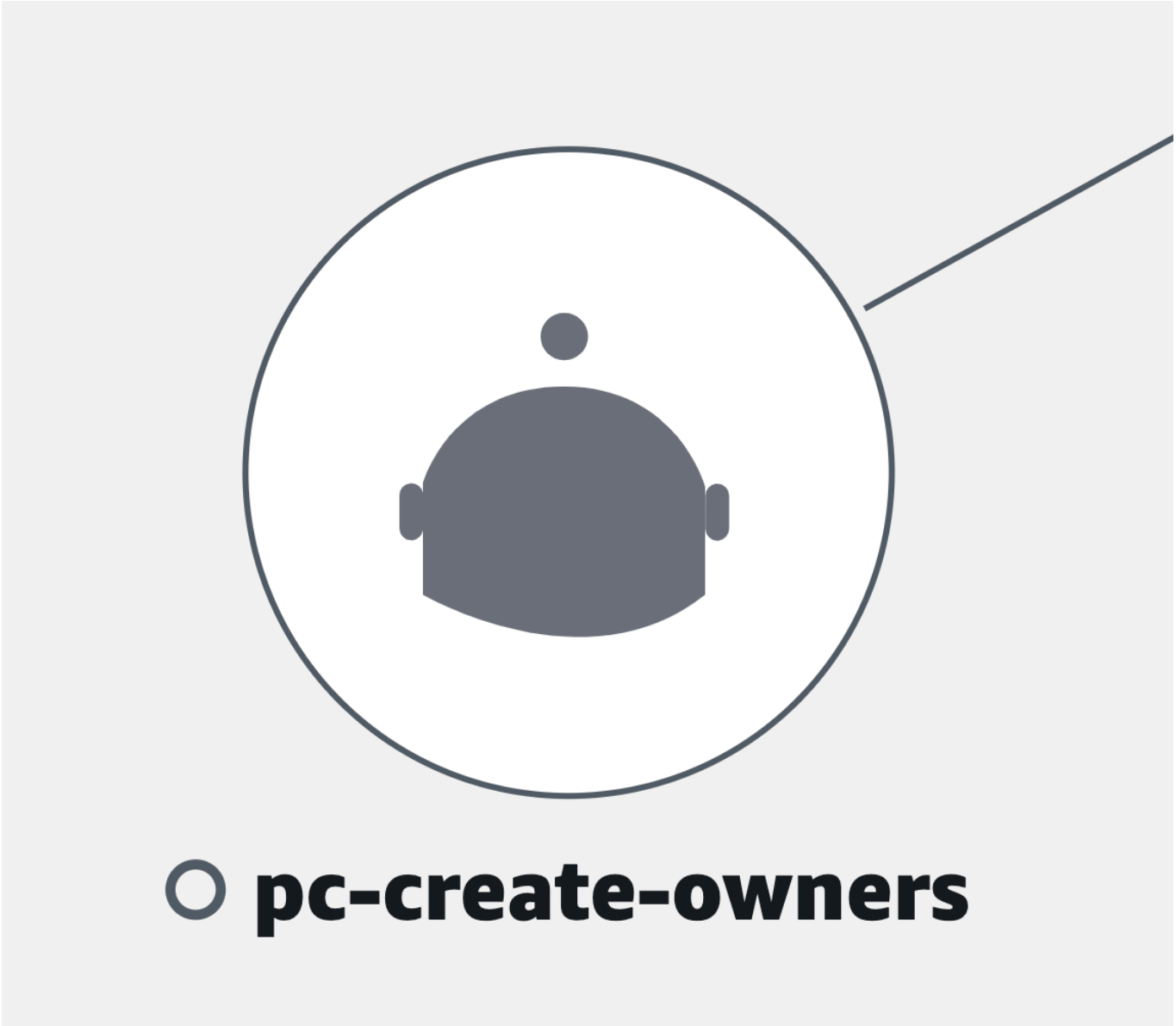
Wählen Sie einen Canary-Node aus, um einen Bereich mit detaillierten Canary-Informationen zu öffnen, wie in der folgenden Abbildung dargestellt:

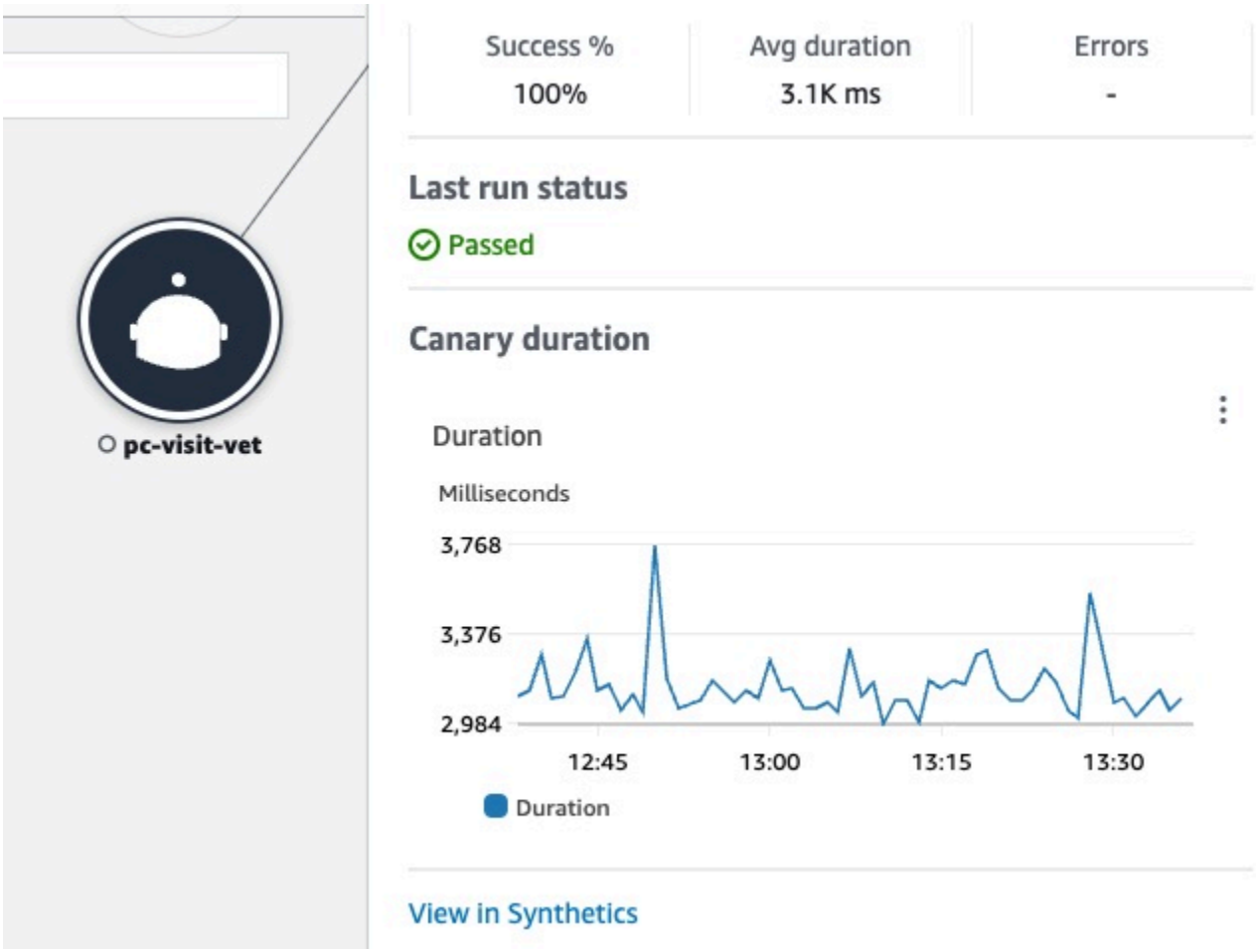


Canaries sind standardmäßig zu einem einzigen erweiterbaren Symbol zusammengefasst. Wählen Sie das Symbol (+), wie im vorherigen Bild gezeigt, um die Gruppe zu erweitern und ihre einzelnen Elemente zu sehen.

Die folgenden Symbole stellen Beispiele für Kunden in der Service Map dar:

- Ein Kanarienvogel aus Kunststoff —


pc-create-owners



Im Bereich für kanarische Knoten können Sie Folgendes sehen:

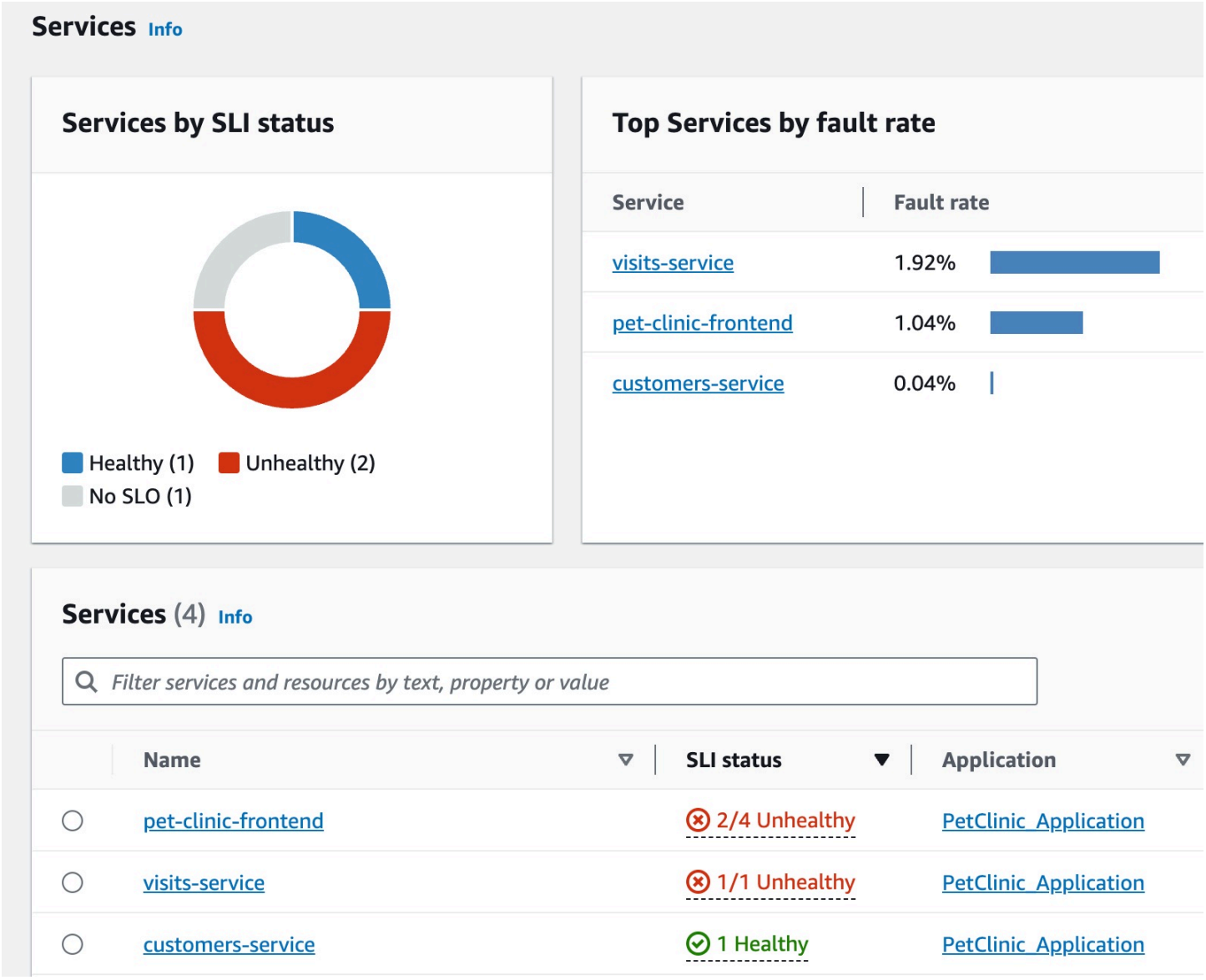
- Metriken für Erfolgsquote, durchschnittliche Dauer und Fehler.
- Der Status der letzten Canary-Ausführung.
- Ein Diagramm, das die Dauer der Canary-Ausführung anzeigt. Zeigen Sie mit der Maus auf eine Grafikserie, um ein Pop-up mit weiteren Informationen zu öffnen.
- Ein Link zur Anzeige kanarischer Details in CloudWatch Synthetics.

Beispiel: Application Signals verwenden, um ein Problem mit dem Betriebsstatus zu beheben

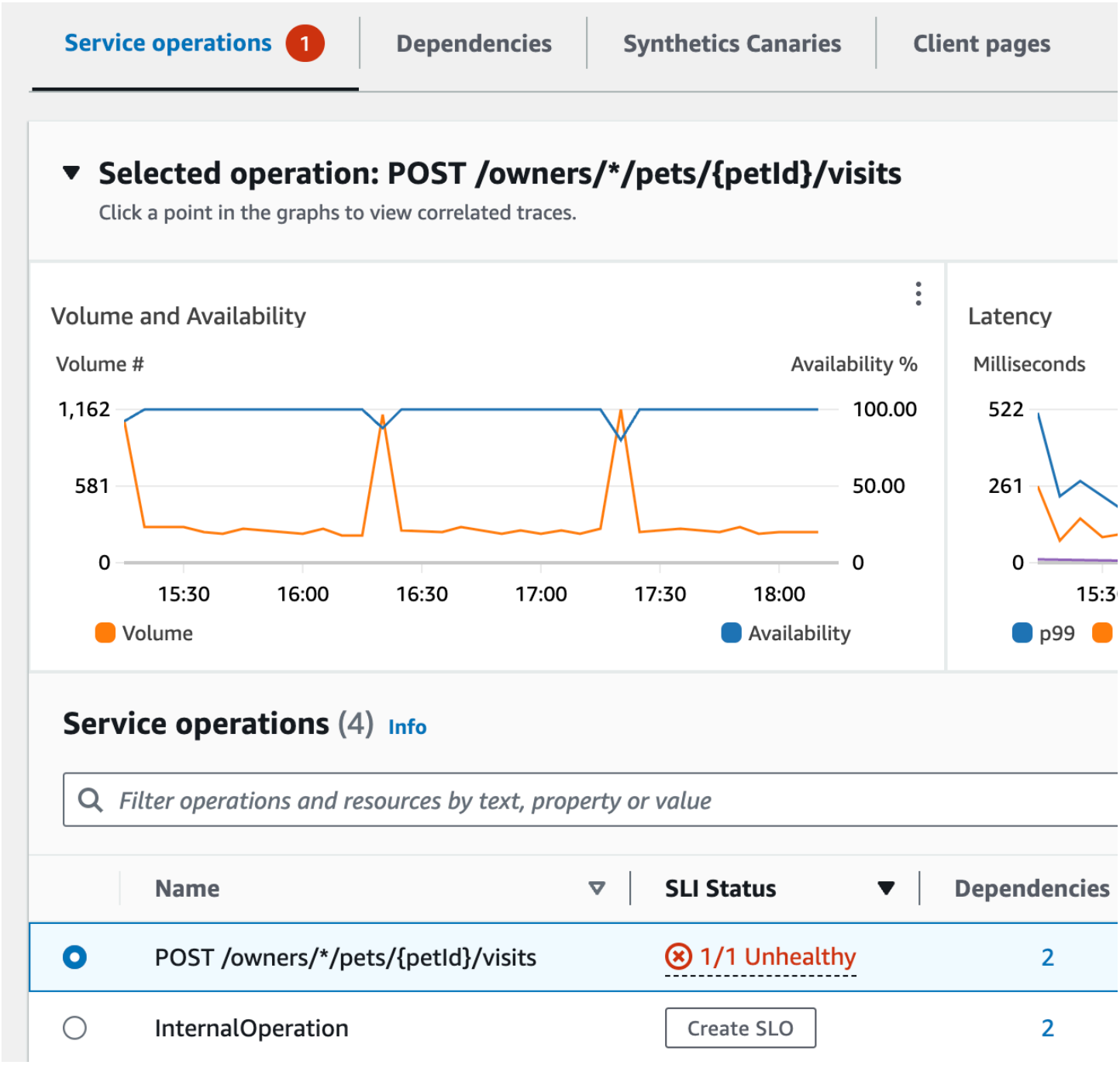
⚠ Application Signals befindet sich in der Vorschauversion für Amazon CloudWatch und kann sich ändern.

Das folgende Szenario bietet ein Beispiel dafür, wie Application Signals verwendet werden kann, um Ihre Services zu überwachen und Probleme mit der Servicequalität zu identifizieren. Gehen Sie ins Detail, um mögliche Ursachen zu ermitteln und Maßnahmen zur Behebung des Problems zu ergreifen. Dieses Beispiel konzentriert sich auf eine Anwendung für Tierkliniken, die aus mehreren Microservices besteht, die AWS-Services beispielsweise DynamoDB aufrufen.

Jane ist Teil eines DevOps Teams, das für den Betrieb einer Anwendung in einer Tierklinik zuständig ist. Janes Team setzt sich dafür ein, dass die Anwendung hochverfügbar und reaktionsschnell ist. Sie verwenden Servicelevel-Ziele (SLOs), um die Anwendungsleistung anhand dieser geschäftlichen Verpflichtungen zu messen. Sie erhält eine Warnung über mehrere fehlerhafte Servicelevel-Indikatoren (SLIs). Sie öffnet die CloudWatch Konsole und navigiert zur Seite Dienste, auf der sie mehrere Dienste sieht, die sich in einem fehlerhaften Zustand befinden.

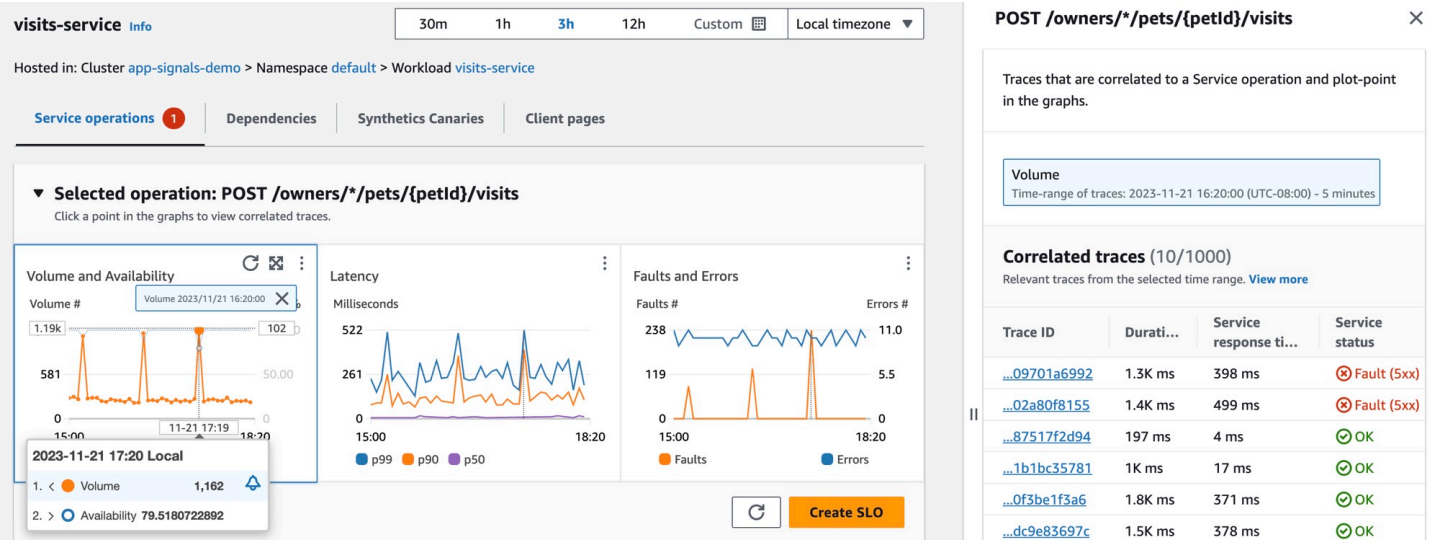


Oben auf der Seite sieht Jane, dass `visits-service` der Service mit der höchsten Fehlerrate ist. Sie wählt den Link im Diagramm aus, wodurch die Seite mit den Service-Details für den Service geöffnet wird. Sie stellt fest, dass in der Tabelle mit den Service-Vorgängen ein fehlerhafter Vorgang vorliegt. Sie wählt diesen Vorgang aus und sieht im Volumen- und Verfügbarkeitsdiagramm, dass es periodische Spitzen im Aufrufvolumen gibt, die mit Verfügbarkeitseinbrüchen zu korrelieren scheinen.

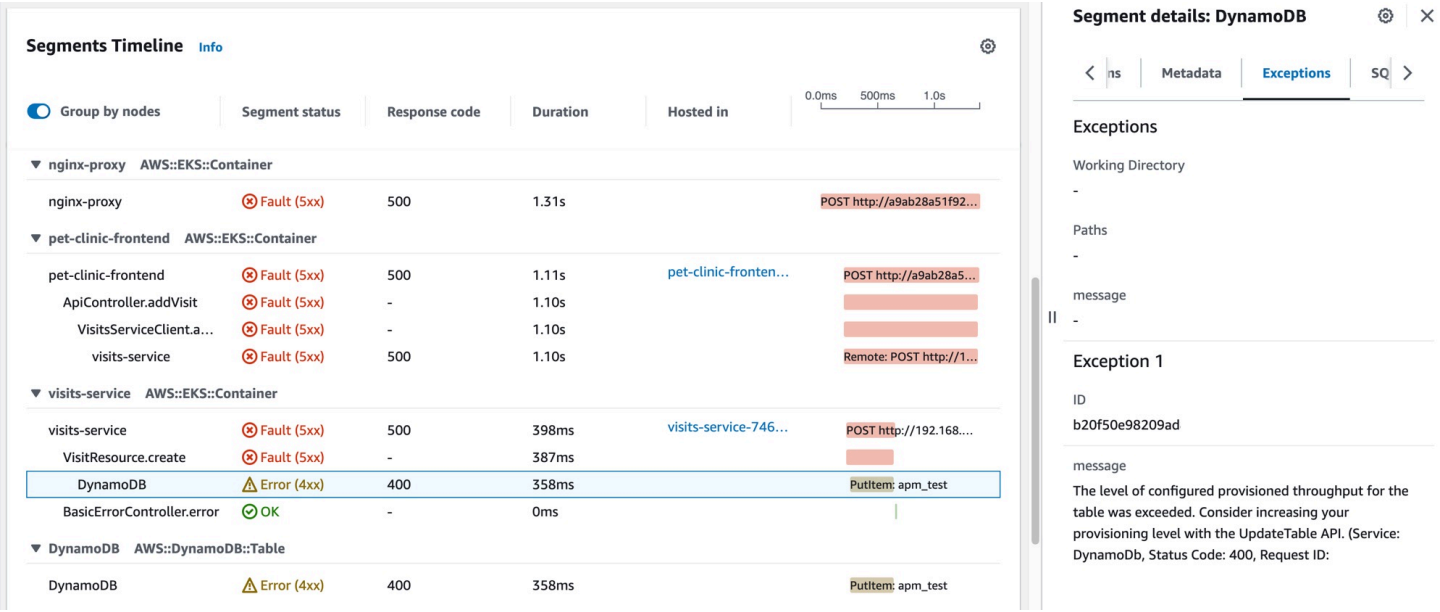


Um sich die Einbrüche der Serviceverfügbarkeit genauer anzusehen, wählt Jane einen der Verfügbarkeits-Datenpunkte im Diagramm aus. Es öffnet sich eine Leiste mit X-Ray-Traces, die mit

dem ausgewählten Datenpunkt korreliert sind. Sie sieht, dass es mehrere Traces gibt, die Fehler enthalten.



Jane wählt eine der korrelierten Traces mit einem Fehlerstatus aus, wodurch die X-Ray-Trace-Detailseite für das ausgewählte Trace geöffnet wird. Jane scrollt nach unten zum Abschnitt Segment-Timeline und folgt dem Aufrufpfad, bis sie feststellt, dass Aufrufe einer DynamoDB-Tabelle Fehler zurückgeben. Sie wählt das DynamoDB-Segment aus und navigiert zur Ausnahmen-Registerkarte in der rechten Leiste.



Jane stellt fest, dass eine DynamoDB-Ressource falsch konfiguriert ist, was bei hohen Kundenanfragen zu Fehlern führt. Der bereitgestellte Durchsatz der DynamoDB-Tabelle wird regelmäßig überschritten, was zu Problemen mit der Service-Verfügbarkeit und fehlerhaften SLIs führt. Auf der Grundlage dieser Informationen ist ihr Team in der Lage, einen höheren

Bereitstellungsdurchsatz zu konfigurieren und eine hohe Verfügbarkeit der Anwendung sicherzustellen.

Erfasste Standard-Anwendungsmetriken

⚠ Application Signals befindet sich in der Vorschauversion. Wenn Sie Feedback zu dieser Funktion haben, können Sie uns unter app-signals-feedback@amazon .com kontaktieren.

Application Signals erfasst Standard-Anwendungsmetriken von den Services, die es entdeckt. Diese Metriken beziehen sich auf die wichtigsten Aspekte der Leistung eines Services: Latenz, Störungen und Fehler. Diese können Ihnen helfen, Probleme zu identifizieren, Leistungstrends zu überwachen und Ressourcen zu optimieren, um das allgemeine Benutzererlebnis zu verbessern.

Die folgende Tabelle listet die Metriken auf, die von Application Signals erfasst werden. Diese Metriken werden CloudWatch an den `AppSignals` Namespace gesendet.

Metrik	Beschreibung
`Latency`	Die Verzögerung vor der Datenübertragung beginnt, nachdem die Anfrage gestellt wurde. Einheiten: Millisekunden
`Faults`	Eine Anzahl von serverseitigen HTTP 5XX-Fehlern und OpenTelemetry Span-Status-Fehlern. Einheiten: keine
`Errors`	Eine Anzahl von clientseitigen HTTP-4XX-Fehlern. Dabei handelt es sich um Anforderungsfehler, die nicht durch Serviceprobleme verursach t werden. Daher betrachtet die auf den Dashboards von Application Signals angezeigte `Availability` -Metrik diese Fehler nicht als Servicefehler. Einheiten: keine

Die auf den Dashboards von Application Signals angezeigte `Availability` Metrik wird als (1 — **`Faults`** /Gesamt) *100 berechnet. Erfolgreiche Antworten sind alle Antworten ohne einem 5XX-Fehler. 4XX-Antworten werden als erfolgreich behandelt, wenn Application Signals `Availability` berechnet.

Erfasste Dimensionen und Dimensionskombinationen

Die folgenden Dimensionen sind für jede der Standard-Anwendungsmetriken definiert. Weitere Informationen zu Dimensionen finden Sie unter Dimensionen.

Für Service-Metriken und Abhängigkeitsmetriken werden unterschiedliche Dimensionen erfasst. Wenn innerhalb der von Application Signals erkannten Services Microservice A Microservice B aufruft, wird die Anforderung an Microservice B bereitgestellt. In diesem Fall gibt Microservice A Abhängigkeitsmetriken und Microservice B Service-Metriken aus. Wenn ein Client Microservice A aufruft, stellt Microservice A die Anforderung und gibt Service-Metriken aus.

Dimensionen für Service-Metriken

Die folgenden Dimensionen werden für Service-Metriken erfasst.

Dimension	Beschreibung
`Service`	Der Name des Service.
`Operation`	Der Name des API-Vorgangs oder der anderen Aktivität.
`HostedIn.EKS.Cluster`	Der Name des Amazon-EKS-Clusters, in dem die Services ausgeführt werden. Diese Dimension wird nur erfasst, wenn die Services auf Amazon EKS ausgeführt werden.
`HostedIn.K8s.Namespace`	Name des Kubernetes-Namespace, in dem die Services ausgeführt werden. Diese Dimension wird nur erfasst, wenn die Services auf Amazon EKS ausgeführt werden.
`HostedIn.Environment`	Benutzerdefinierter Name der Umgebung, in der die Services Dienste ausgeführt werden.

Dimension	Beschreibung
	Diese Dimension wird nur erfasst, wenn die Services in einer Umgebung ausgeführt werden, die nicht Amazon EKS ist.

Wenn Sie sich diese Metriken in der CloudWatch Konsole ansehen, können Sie wählen, ob Sie sie mit den folgenden Dimensionskombinationen anzeigen möchten.

- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`

Für Plattformen, die nicht Amazon EKS sind, können Sie auch Service-Metriken mit den folgenden Dimensions-Kombinationen anzeigen.

- `Service`, `Operation`, `HostedIn.Environment`
- `Service`, `HostedIn.Environment`

Dimensionen für Abhängigkeitsmetriken

Die folgenden Dimensionen werden für Abhängigkeitsmetriken erfasst.

Dimension	Beschreibung
`Service`	Der Name des Service.
`Operation`	Der Name des API-Vorgangs oder der anderen Aktivität.
`RemoteService`	Name des aufgerufenen Remote-Services.
`RemoteOperation`	Name des API-Vorgangs, der aufgerufen wird.
`HostedIn.EKS.Cluster`	Der Name des Amazon-EKS-Clusters, in dem die Services ausgeführt werden. Diese Dimension wird nur erfasst, wenn die Services auf Amazon EKS ausgeführt werden.

Dimension	Beschreibung
`HostedIn.K8s.Namespace`	Name des Kubernetes-Namespace, in dem die Services ausgeführt werden. Diese Dimension wird nur erfasst, wenn die Services auf Amazon EKS ausgeführt werden.
`K8s.RemoteNamespace`	Der Name des Kubernetes-Namespace, in dem die abhängigen Services ausgeführt werden. Diese Dimension wird nur erfasst, wenn die Services auf Amazon EKS ausgeführt werden.
`RemoteTarget`	Name der Ressource, die durch die Remote-Aufrufe aufgerufen wird. Diese Dimension hat keinen Wert, wenn die Remote-Aufrufe nicht an bestimmte Ressourcen gerichtet sind. Diese Dimension wird nur erfasst, wenn die Services auf Amazon EKS ausgeführt werden.
`HostedIn.Environment`	Benutzerdefinierter Name der Umgebung, in der die Services Dienste ausgeführt werden. Diese Dimension wird nur erfasst, wenn die Services in einer Umgebung ausgeführt werden, die nicht Amazon EKS ist.

Wenn Sie diese Metriken in der CloudWatch Konsole anzeigen, können Sie wählen, ob Sie sie mit den folgenden Dimensionskombinationen anzeigen möchten.

Wird auf jeder beliebigen Plattform ausgeführt

- `RemoteService`

Wird auf Amazon-EKS-Clustern ausgeführt

- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`, `RemoteTarget`

- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`
- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `RemoteService`, `RemoteOperation`,
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `K8s.RemoteNamespace`
- `Service`, `HostedIn.EKS.Cluster`, `RemoteService`, `K8s.RemoteNamespace`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`, `RemoteTarget`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`

Wird auf anderen Plattformen als Amazon-EKS-Clustern ausgeführt

- `Service`, `Operation`, `HostedIn.Environment`
- `Service`, `HostedIn.Environment`
- `Service`, `Operation`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `Operation`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`,
- `Service`, `HostedIn.Environment`, `RemoteService`
- `Service`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`,

Verwenden Sie synthetisches Monitoring

Sie können Amazon CloudWatch Synthetics verwenden, um kanarische, konfigurierbare Skripts zu erstellen, die nach einem Zeitplan ausgeführt werden, um Ihre Endgeräte und APIs zu überwachen.

Canarys folgen denselben Routen und führen dieselben Aktionen aus wie ein Kunde. So können Sie Ihre Kundenerfahrung kontinuierlich überprüfen, auch wenn Sie keinen Kundenverkehr auf Ihren Anwendungen haben. Durch den Einsatz von Canarys können Sie Probleme entdecken, bevor Ihre Kunden sie entdecken.

Canarys sind Skripts, die in Node.js oder Python geschrieben wurden. Sie legen Lambda-Funktionen in Ihrem Konto an, die Node.js oder Python als Framework verwenden. Canarys arbeiten über HTTP- und HTTPS-Protokolle. Die Kanaren verwenden Lambda-Schichten, die die CloudWatch Synthetics-Bibliothek enthalten. Die Bibliothek enthält die NodeJS-Version von CloudWatch Synthetics for NodeJS Canaries und die Python-Version von Synthetics for Python Canaries. CloudWatch Die Layer gehören zum CloudWatch Synthetics-Dienstkonto. Bibliotheken übertragen oder speichern niemals Kundeninformationen. Alle Kundendaten werden nur im Kundenkonto gespeichert.

Canarys bieten programmatischen Zugriff auf einen Headless-Google-Chrome-Browser über Puppeteer oder Selenium Webdriver. Weitere Informationen zu Puppeteer finden Sie unter Puppeteer. Weitere Informationen zu Selenium finden Sie unter www.selenium.dev/.

Canarys überprüfen die Verfügbarkeit und Latenz Ihrer Endpunkte und können Daten zur Ladezeit und Screenshots der Benutzeroberfläche speichern. Sie überwachen Ihre REST-APIs, URLs und Website-Inhalte und können auf nicht autorisierte Änderungen durch Phishing, Code-Injection und Cross-Site-Scripting prüfen.

CloudWatch Synthetics ist in Application Signals integriert, das Ihre Anwendungsdienste, Clients, Synthetics-Kanarien und Serviceabhängigkeiten erkennen und überwachen kann. Verwenden Sie Application Signals, um eine Liste oder eine visuelle Übersicht Ihrer Services zu erhalten, Zustandsmetriken auf der Grundlage Ihrer Servicelevel-Ziele (SLOs) einzusehen und eine detaillierte Darstellung korrelierter X-Ray-Traces für eine detailliertere Fehlerbehebung durchzuführen. Um Ihre Canarys in Application Signals zu sehen, aktivieren Sie X-Ray Active Tracing. Ihre Canarys werden auf der Service-Karte angezeigt, die mit Ihren Services verbunden ist, sowie auf der Service-Detailseite der Services, die sie aufrufen.

Eine Video-Demonstration von Canarys finden Sie hier:

- Einführung in Amazon CloudWatch Synthetics
- Amazon CloudWatch Synthetics Demo
- Erstellen Sie Kanarienvögel mit Amazon Synthetics CloudWatch
- Visuelle Überwachung mit Amazon CloudWatch Synthetics

Sie können ein Canary einmal oder nach einem regelmäßigen Zeitplan ausführen. Canarys können bis zu einmal pro Minute laufen. Sie können sowohl Cron- als auch Rate-Ausdrücke verwenden, um Canarys zu planen.

Informationen zu Sicherheitsproblemen, die vor dem Erstellen und Ausführen von Canarys berücksichtigt werden sollten, finden Sie unter Sicherheitsüberlegungen für Synthetics-Canaries.

Standardmäßig erstellen Canaries mehrere CloudWatch Metriken im `CloudWatchSynthetics` Namespace. Diese Metriken erhalten `CanaryName` als Dimension. Canarys, die die `executeStep()`- oder `executeHttpStep()`-Funktion aus der Funktionsbibliothek verwenden, erhalten ebenfalls `StepName` als Dimension. Weitere Informationen zur Canary-Funktionsbibliothek finden Sie unter Für Canary-Skripte verfügbare Bibliotheksfunktionen.

CloudWatch Synthetics lässt sich gut in die X-Ray Trace Map integrieren, AWS X-Ray die CloudWatch mit einen end-to-end Überblick über Ihre Dienste bietet, sodass Sie Leistungsengpässe effizienter erkennen und betroffene Benutzer identifizieren können. Kanarienvögel, die Sie mit CloudWatch Synthetics erstellen, werden auf der Trace-Map angezeigt. Weitere Informationen finden Sie unter X-Ray Trace Map.

CloudWatch Synthetics ist derzeit in allen AWS Handelsregionen und Regionen erhältlich. GovCloud

> **Note**
>
> Im asiatisch-pazifischen Raum (Osaka) AWS PrivateLink wird dies nicht unterstützt. In der Region Asien-Pazifik (Jakarta) werden AWS PrivateLink und X-Ray nicht unterstützt.

Themen

- Anzeigen von Canary-Statistiken und -Details
- CloudWatch von Canaries veröffentlichte Metriken
- Einen Canary bearbeiten oder löschen
- Laufzeit für mehrere Canary starten, stoppen, löschen oder aktualisieren
- Überwachung kanarischer Ereignisse mit Amazon EventBridge

Erforderliche Rollen und Berechtigungen für CloudWatch Kanarienvögel

Sowohl die Benutzer, die Canarys erstellen und verwalten, als auch die Canarys selbst benötigen bestimmte Berechtigungen.

Erforderliche Rollen und Berechtigungen für Benutzer, die Canaries verwalten CloudWatch

Um Canary-Details und die Ergebnisse von Canary-Ausführungen anzuzeigen, müssen Sie als Benutzer mit `CloudWatchSyntheticsFullAccess` oder `CloudWatchSyntheticsReadOnlyAccess` und den angehängten Richtlinien angemeldet sein. Um alle Synthetics-Daten in der Konsole lesen zu können, benötigen Sie außerdem die Richtlinien `AmazonS3ReadOnlyAccess` und `CloudWatchReadOnlyAccess`. Um den von Canarys verwendeten Quellcode anzeigen zu können, benötigen Sie auch die `AWSLambda_ReadOnlyAccess`-Richtlinie.

Um Canarys erstellen zu können, müssen Sie als ein Benutzer angemeldet sein, der über die `CloudWatchSyntheticsFullAccess`-Richtlinie oder einen ähnlichen Satz von Berechtigungen verfügt. Um IAM-Rollen für die Canarys erstellen zu können, benötigen Sie außerdem die folgende Inline-Richtlinienanweisung:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:CreatePolicy",
                "iam:AttachRolePolicy"
            ],
            "Resource": [
```

```
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*",
                "arn:aws:iam::*:policy/service-role/CloudWatchSyntheticsPolicy*"
            ]
        }
    ]
}
```

> **Important**
>
> Wenn Sie einem Benutzer die `iam:AttachRolePolicy` Berechtigungen `iam:CreateRoleiam:CreatePolicy`, und gewähren, erhält dieser Benutzer vollen Administratorzugriff auf Ihr AWS Konto. Beispielsweise kann ein Benutzer mit diesen Berechtigungen eine Richtlinie erstellen, die über vollständige Berechtigungen für alle Ressourcen verfügt, und diese Richtlinie an eine beliebige Rolle anhängen. Seien Sie sehr vorsichtig, wem Sie diese Berechtigungen erteilen.

Informationen zum Anhängen von Richtlinien und zum Erteilen von Berechtigungen für Benutzer finden Sie unter Ändern von Berechtigungen für einen IAM-Benutzer und So betten Sie eine Inline-Richtlinie für einen Benutzer oder eine Rolle ein.

Erforderliche Rollen und Berechtigungen für Canarys

Jedem Canary muss eine IAM-Rolle zugeordnet sein, der bestimmte Berechtigungen zugewiesen sind. Wenn Sie mit der CloudWatch Konsole einen Canary erstellen, können Sie CloudWatch Synthetics wählen, um eine IAM-Rolle für den Canary zu erstellen. Wenn Sie dies tun, verfügt die Rolle über die erforderlichen Berechtigungen.

Wenn Sie die IAM-Rolle selbst erstellen oder eine IAM-Rolle erstellen möchten, die Sie bei Verwendung der AWS CLI oder von APIs zum Erstellen eines Canary nutzen können, muss die Rolle über die in diesem Abschnitt aufgeführten Berechtigungen verfügen.

Alle IAM-Rollen für Canarys müssen die folgende Vertrauensrichtlinienanweisung enthalten.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
```

```
                "Service": "lambda.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Darüber hinaus muss die IAM-Rolle des Canary eine der folgenden Anweisungen enthalten.

Basic Canary, das keinen Amazon VPC-Zugriff verwendet AWS KMS oder benötigt

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/s3/bucket/canary/results/folder"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/s3/bucket/that/contains/canary/results"

            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
               "StringEquals": {
                "cloudwatch:namespace": "CloudWatchSynthetics"
               }
            }
        }
    ]
}
```

Canary, das AWS KMS zur Verschlüsselung kanarischer Artefakte verwendet wird, aber keinen Amazon VPC-Zugriff benötigt

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/S3/bucket/canary/results/folder"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/S3/bucket/that/contains/canary/results"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CloudWatchSynthetics"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
```

```
            "Resource":
 "arn:aws:kms:KMS_key_region_name:KMS_key_account_id:key/KMS_key_id",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "s3.region_name_of_the_canary_results_S3_bucket.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

Canary, das Amazon VPC-Zugriff nicht verwendet, AWS KMS aber benötigt

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/S3/bucket/canary/results/folder"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/S3/bucket/that/contains/canary/results"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
```

```
            "Resource": [
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CloudWatchSynthetics"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": [
                "*"
            ]
       }
    ]
}
```

Canary, AWS KMS das kanarische Artefakte verschlüsselt und außerdem Amazon VPC-Zugriff benötigt

Wenn Sie einen Nicht-VPC-Canary aktualisieren, um mit der Verwendung einer VPC zu beginnen, müssen Sie die Rolle des Canary aktualisieren, um die in der folgenden Richtlinie aufgeführten Netzwerkschnittstellenberechtigungen aufzunehmen.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/S3/bucket/canary/results/folder"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/S3/bucket/that/contains/canary/results"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
            ],
```

```
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CloudWatchSynthetics"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": [
                "*"
            ]
       },
       {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource":
 "arn:aws:kms:KMS_key_region_name:KMS_key_account_id:key/KMS_key_id",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "s3.region_name_of_the_canary_results_S3_bucket.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

AWS verwaltete Richtlinien für CloudWatch Synthetics

Um Benutzern, Gruppen und Rollen Berechtigungen hinzuzufügen, ist es einfacher, von AWS verwaltete Richtlinien zu verwenden, als selbst Richtlinien zu schreiben. Es erfordert Zeit und Fachwissen, um von Kunden verwaltete IAM-Richtlinien zu erstellen, die Ihrem Team nur die benötigten Berechtigungen bieten. Um schnell loszulegen, können Sie unsere AWS verwalteten Richtlinien verwenden. Diese Richtlinien decken allgemeine Anwendungsfälle ab und sind in Ihrem AWS Konto verfügbar. Weitere Informationen zu AWS -verwalteten Richtlinien finden Sie unter AWS managed Policies AWS -verwaltete Richtlinien im IAM-Benutzerhandbuch.

AWS Dienste verwalten und aktualisieren AWS verwaltete Richtlinien. Sie können die Berechtigungen in AWS verwalteten Richtlinien nicht ändern. Dienste ändern gelegentlich die Berechtigungen in einer AWS -verwalteten Richtlinie. Diese Art von Update betrifft alle Identitäten (Benutzer, Gruppen und Rollen), an welche die Richtlinie angehängt ist.

CloudWatch Synthetics-Updates für AWS verwaltete Richtlinien

Sehen Sie sich Details zu Aktualisierungen der AWS verwalteten Richtlinien für CloudWatch Synthetics an, seit dieser Dienst begonnen hat, diese Änderungen zu verfolgen. Abonnieren Sie den RSS-Feed auf der Seite CloudWatch Dokumentenverlauf, um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten.

Änderung	Beschreibung	Datum
Redundante Aktionen wurden entfernt von CloudWatchSyntheticsFullAccess	CloudWatch Synthetics hat die `lambda:GetLayerVersionByArn` Aktionen `s3:PutBucketEncryption` und aus der CloudWatchSyntheticsFullAccessRichtlinie entfernt, da diese Aktionen mit anderen Berechtigungen in der Richtlinie überflüssig waren. Die entfernten Aktionen stellten keine Berechtigungen bereit, und es gibt keine Nettoänderung an	12. März 2021

Änderung	Beschreibung	Datum
	den Berechtigungen, die von der Richtlinie gewährt werden.	
CloudWatch Synthetics begann, Änderungen zu verfolgen	CloudWatch Synthetics begann, Änderungen für seine AWS verwalteten Richtlinien zu verfolgen.	10. März 2021

CloudWatchSyntheticsFullAccess

Hier sind die Inhalte der `CloudWatchSyntheticsFullAccess`-Richtlinie:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "synthetics:*"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:CreateBucket",
            "s3:PutEncryptionConfiguration"
         ],
         "Resource":[
            "arn:aws:s3:::cw-syn-results-*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:ListRoles",
            "s3:ListAllMyBuckets",
            "s3:GetBucketLocation",
            "xray:GetTraceSummaries",
            "xray:BatchGetTraces",
```

```
            "apigateway:GET"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:ListBucket"
         ],
         "Resource":"arn:aws:s3:::cw-syn-*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersion"
         ],
         "Resource":"arn:aws:s3:::aws-synthetics-library-*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:PassRole"
         ],
         "Resource":[
            "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
         ],
         "Condition":{
            "StringEquals":{
               "iam:PassedToService":[
                  "lambda.amazonaws.com",
                  "synthetics.amazonaws.com"
               ]
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:GetRole"
         ],
         "Resource":[
            "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
         ]
```

```
      },
      {
         "Effect":"Allow",
         "Action":[
            "cloudwatch:GetMetricData",
            "cloudwatch:GetMetricStatistics"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "cloudwatch:PutMetricAlarm",
            "cloudwatch:DeleteAlarms"
         ],
         "Resource":[
            "arn:aws:cloudwatch:*:*:alarm:Synthetics-*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "cloudwatch:DescribeAlarms"
         ],
         "Resource":[
            "arn:aws:cloudwatch:*:*:alarm:*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "lambda:CreateFunction",
            "lambda:AddPermission",
            "lambda:PublishVersion",
            "lambda:UpdateFunctionConfiguration",
            "lambda:GetFunctionConfiguration"
         ],
         "Resource":[
            "arn:aws:lambda:*:*:function:cwsyn-*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
```

```
            "lambda:GetLayerVersion",
            "lambda:PublishLayerVersion"
         ],
         "Resource":[
            "arn:aws:lambda:*:*:layer:cwsyn-*",
            "arn:aws:lambda:*:*:layer:Synthetics:*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "ec2:DescribeVpcs",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups"
         ],
         "Resource":[
            "*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "sns:ListTopics"
         ],
         "Resource":[
            "*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "sns:CreateTopic",
            "sns:Subscribe",
            "sns:ListSubscriptionsByTopic"
         ],
         "Resource":[
            "arn:*:sns:*:*:Synthetics-*"
         ]
      }
   ]
}
```

CloudWatchSyntheticsReadOnlyAccess

Hier sind die Inhalte der `CloudWatchSyntheticsReadOnlyAccess`-Richtlinie:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:Describe*",
                "synthetics:Get*",
                "synthetics:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

Einen Benutzer auf das Anzeigen bestimmter canarys beschränken

Sie können die Fähigkeit eines Benutzers einschränken, Informationen über canarys anzuzeigen, sodass er nur Informationen über die von Ihnen angegebenen canarys sehen kann. Verwenden Sie dazu eine IAM-Richtlinie mit einer `Condition`-Anweisung ähnlich der folgenden, und fügen Sie diese Richtlinie an einen Benutzer oder eine IAM-Rolle an.

Im folgenden Beispiel wird der Benutzer darauf beschränkt, nur Informationen über `name-of-allowed-canary-1` und `name-of-allowed-canary-2` anzuzeigen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "synthetics:DescribeCanaries",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "synthetics:Names": [
                        "name-of-allowed-canary-1",
                        "name-of-allowed-canary-2"
                    ]
                }
            }
```

```
        }
    ]
}
```

CloudWatch Synthetics unterstützt das Auflisten von bis zu fünf Elementen im `synthetics:Names` Array.

Sie können auch eine Richtlinie erstellen, die ein * als Platzhalter in canary-Namen verwendet, die zulässig sein sollen, wie im folgenden Beispiel:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "synthetics:DescribeCanaries",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringLike": {
                    "synthetics:Names": [
                        "my-team-canary-*"
                    ]
                }
            }
        }
    ]
}
```

Jeder Benutzer, der mit einer dieser Richtlinien angemeldet ist, kann die CloudWatch Konsole nicht zum Anzeigen von Canary-Informationen verwenden. Sie können Informationen zu Canaries nur für die Kanarischen Inseln einsehen, die durch die Richtlinie autorisiert wurden, und nur, wenn sie die DescribeCanariesAPI oder den Befehl AWS CLI describe-canaries verwenden.

Erstellen eines Canarys

> **Important**
>
> Stellen Sie sicher, dass Sie Synthetics-Canaries verwenden, um nur Endpunkte und APIs zu überwachen, für die Sie die Eigentümerschaft oder Berechtigungen haben. Abhängig von

den Einstellungen für die Canary-Frequenz kann es bei diesen Endpunkten ggf. zu einem erhöhten Datenverkehr kommen.

Wenn Sie die CloudWatch Konsole verwenden, um einen Canary zu erstellen, können Sie einen von bereitgestellten Blueprint verwenden, um Ihren Canary CloudWatch zu erstellen, oder Sie können Ihr eigenes Skript schreiben. Weitere Informationen finden Sie unter Verwenden von Canary-Vorlagen.

Sie können auch einen Canary erstellen, AWS CloudFormation wenn Sie Ihr eigenes Skript für den Canary verwenden. Weitere Informationen finden Sie AWS::Synthetics::Canaryim AWS CloudFormation Benutzerhandbuch.

Wenn Sie Ihr eigenes Skript schreiben, können Sie mehrere Funktionen verwenden, die CloudWatch Synthetics in eine Bibliothek integriert hat. Weitere Informationen finden Sie unter Synthetics Laufzeitversionen.

> **Note**
>
> Wenn Sie einen Kanarienvogel erstellen, handelt es sich bei einer der erstellten Ebenen um eine Synthetics-Ebene, der vorangestellt ist. `Synthetics` Diese Ebene gehört dem Synthetics-Dienstkonto und enthält den Laufzeitcode.

So erstellen Sie ein Canary

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Synthetics Canaries aus.
3. Wählen Sie Canary erstellen aus.
4. Wählen Sie eine der folgenden Optionen aus:
   - Um Ihr Canary auf der Grundlage eines Vorlagenskripts zu erstellen, wählen Sie die Option Use a blueprint (Eine Vorlage verwenden), und wählen Sie dann die Art von Canary aus, das Sie erstellen möchten. Weitere Informationen darüber, was jeder Typ von Vorlage bewirkt, finden Sie unter Verwenden von Canary-Vorlagen.
   - Um ein eigenes Node.js-Skript zum Erstellen eines benutzerdefinierten Canarys hochzuladen, wählen Sie Upload a script (Skript hochladen) aus.

Sie können dann Ihr Skript in den Script (Skript)-Bereich ziehen oder Browse files (Dateien durchsuchen) auswählen, um zum Skript in Ihrem Dateisystem zu navigieren.

- Um das Skript aus einem S3-Bucket zu importieren, wählen Sie Import from S3 (Aus S3 importieren) aus. Geben Sie unter Source location (Quellspeicherort) den vollständigen Pfad zu Ihrem Canary ein oder wählen Sie Browse S3 (S3 durchsuchen) aus.

  Sie müssen über `s3:GetObject`- und `s3:GetObjectVersion`-Berechtigungen für den S3-Bucket verfügen, den Sie verwenden. Der Bucket muss sich in derselben AWS Region befinden, in der Sie den Canary erstellen.

5. Geben Sie unter Name einen Namen für Ihr Canary ein. Der Name wird auf vielen Seiten verwendet, daher empfehlen wir Ihnen, ihm einen beschreibenden Namen zu geben, der es von anderen Canaries unterscheidet.
6. Geben Sie unter Application or endpoint URL (Anwendungs- oder Endpunkt-URL) die URL ein, die das Canary testen soll. Diese URL muss das Protokoll enthalten (z. B. https://).

   Wenn das Canary einen Endpunkt auf einer VPC testen soll, müssen Sie später in diesem Verfahren auch Informationen zu Ihrer VPC eingeben.
7. Wenn Sie ein eigenes Skript für das Canary verwenden, geben Sie unter Lambda-Handler den Einstiegspunkt ein, an dem das Canary beginnen soll. Wenn Sie eine Laufzeit vor `syn-nodejs-puppeteer-3.4` oder `syn-python-selenium-1.1` verwenden, muss die von Ihnen eingegebene Zeichenfolge auf `.handler` enden. Wenn Sie `syn-nodejs-puppeteer-3.4` bzw. `syn-python-selenium-1.1` oder eine spätere Laufzeit verwenden, gilt diese Einschränkung nicht.
8. Wenn Sie in Ihrem Skript Umgebungsvariablen verwenden, wählen Sie Umgebungsvariablen und geben Sie dann einen Wert für jede in Ihrem Skript definierte Umgebungsvariable an. Weitere Informationen finden Sie unter Umgebungsvariablen.
9. Wählen Sie unter Zeitplan aus, ob dieser Canary nur einmal, kontinuierlich mit einem Ratenausdruck oder mit einem Cron-Ausdruck geplant werden soll.

   - Wenn Sie die CloudWatch Konsole verwenden, um einen Canary zu erstellen, der kontinuierlich läuft, können Sie eine Rate zwischen einmal pro Minute und einmal pro Stunde wählen.

   - Weitere Informationen zum Schreiben eines Cron-Ausdrucks für Canary-Scheduling finden Sie unter Planen von Canary-Durchläufen mit Cron.

10. (Optional) Um einen Timeout-Wert für den canary festzulegen, wählen Sie Zusätzliche Konfiguration und geben Sie dann den Timeout-Wert an. Legen Sie ihn nicht kürzer als 15 Sekunden fest, um Lambda-Kaltstarts und die Zeit zu ermöglichen, die zum Hochfahren der canary-Instrumentierung benötigt wird.
11. Geben Sie unter Datenaufbewahrung an, wie lange Informationen über fehlgeschlagene und erfolgreiche Canary-Ausführungen aufbewahrt werden sollen. Der Bereich liegt zwischen 1 und 455 Tagen.

    Diese Einstellung wirkt sich nur auf die Daten aus, die CloudWatch Synthetics in der Konsole speichert und anzeigt. Dies hat keinen Einfluss auf die in Ihren Amazon-S3-Buckets gespeicherten Daten oder Protokolle oder Metriken, die vom Canary veröffentlicht werden.
12. Wählen Sie unter Data Storage (Datenspeicher) den S3-Bucket aus, der zum Speichern der Daten aus den Canary-Ausführungen verwendet werden soll. Der Bucket-Name darf keinen Punkt (.) enthalten. Wenn Sie dieses Feld leer lassen, wird ein Standard-S3-Bucket verwendet oder erstellt.

    Wenn Sie die Laufzeit `syn-nodejs-puppeteer-3.0` oder höher verwenden und die URL für den Bucket in das Textfeld eingeben, können Sie einen Bucket in der aktuellen Region oder in einer anderen Region angeben. Wenn Sie eine frühere Laufzeitversion verwenden, muss sich der Bucket in der aktuellen Region befinden.
13. (Optional) Standardmäßig speichern die Kanaren ihre Artefakte auf Amazon S3, und die Artefakte werden im Ruhezustand mit einem AWS verwalteten AWS KMS Schlüssel verschlüsselt. Sie können eine andere Verschlüsselungsoption verwenden, indem Sie Additional configuration (Zusätzliche Konfiguration) im Abschnitt Data Storage (Datenspeicher) auswählen. Sie können dann den Schlüsseltyp auswählen, der für die Verschlüsselung verwendet werden soll. Weitere Informationen finden Sie unter Verschlüsseln von Canary-Artefakten.
14. Wählen Sie unter Access permissions (Zugriffsberechtigungen) aus, ob eine IAM-Rolle zum Ausführen des Canarys erstellt oder eine vorhandene Rolle verwendet werden soll.

    Wenn Sie CloudWatch Synthetics die Rolle erstellen lassen, enthält sie automatisch alle erforderlichen Berechtigungen. Wenn Sie die Rolle selbst erstellen möchten, finden Sie unter Erforderliche Rollen und Berechtigungen für Canarys Informationen zu den erforderlichen Berechtigungen.

    Wenn Sie beim Erstellen des Canaries die CloudWatch Konsole verwenden, um eine Rolle für einen Canary zu erstellen, können Sie die Rolle nicht für andere Canaries wiederverwenden, da

diese Rollen nur für einen Canary spezifisch sind. Wenn Sie eine Rolle manuell erstellt haben, die für mehrere Canarys geeignet ist, können Sie diese vorhandene Rolle verwenden.

Um eine vorhandene Rolle zu verwenden, müssen Sie über die `iam:PassRole`-Berechtigung verfügen, um diese Rolle an Synthetics und Lambda übergeben zu können. Darüber hinaus müssen Sie über die `iam:GetRole`-Berechtigung verfügen.

15. (Optional) Wählen Sie unter Alarme aus, ob CloudWatch Standardalarme für diesen Canary erstellt werden sollen. Wenn Sie Alarme erstellen möchten, werden diese mit der folgenden Namenskonvention erstellt: `Synthetics-Alarm-`*`canaryName`*`-`*`index`*

    `index` ist eine Zahl, die jeden unterschiedlichen Alarm darstellt, der für diesen Canary erstellt wurde. Der erste Alarm hat einen Index von 1, der zweite Alarm hat einen Index von 2 usw.

16. (Optional) Um dieses Canary einen Endpunkt testen zu lassen, der sich auf einer VPC befindet, wählen Sie VPC settings (VPC-Einstellungen) und gehen dann wie folgt vor:

    a. Wählen Sie die VPC aus, die den Endpunkt hostet.

    b. Wählen Sie ein oder mehrere Subnetze auf Ihrer VPC aus. Sie müssen ein privates Subnetz auswählen, da eine Lambda-Instance nicht so konfiguriert werden kann, dass sie in einem öffentlichen Subnetz ausgeführt wird, wenn der Lambda-Instance während der Ausführung keine IP-Adresse zugewiesen werden kann. Weitere Informationen finden Sie unter Konfigurieren einer Lambda-Funktion für den Zugriff auf Ressourcen in einer VPC.

    c. Wählen Sie eine oder mehrere Sicherheitsgruppen auf Ihrer VPC aus.

    Wenn sich der Endpunkt auf einer VPC befindet, müssen Sie Ihren Canary so einrichten, dass er Informationen an CloudWatch Amazon S3 senden kann. Weitere Informationen finden Sie unter Ausführen eines Canarys in einer VPC.

17. (Optional) Fügen Sie unter Tags ein oder mehrere Schlüssel/Wert-Paare als Tags für dieses Canary hinzu. Mithilfe von Tags können Sie Ihre AWS Ressourcen identifizieren und organisieren und Ihre AWS Kosten verfolgen. Weitere Informationen finden Sie unter Verschlagworten Sie Ihre Amazon-Ressourcen CloudWatch .

18. (Optional) Wählen Sie unter Aktive Verfolgung aus, ob die aktive X-Ray-Verfolgung für diesen Canary aktiviert werden soll. Diese Option ist nur verfügbar, wenn der Canary die Laufzeitversion syn-nodejs-2.0 oder höher verwendet. Weitere Informationen finden Sie unter Canary- und X-Ray-Ablaufverfolgung.

Ressourcen, die für Canaries erstellt werden

Wenn Sie ein Canary erstellen, werden die folgenden Ressourcen erstellt:

- Eine IAM-Rolle mit dem Namen `CloudWatchSyntheticsRole-`*`canary-name`*`-`*`uuid`* (wenn Sie die CloudWatch Konsole verwenden, um den Canary zu erstellen und angeben, dass eine neue Rolle für den Canary erstellt werden soll)
- Eine IAM-Richtlinie mit dem Namen `CloudWatchSyntheticsPolicy-`*`canary-name`*`-`*`uuid`*.
- Ein S3-Bucket mit dem Namen `cw-syn-results-`*`accountID`*`-`*`region`*.
- Alarme mit dem Namen `Synthetics-Alarm-`*`MyCanaryName`*, wenn Alarme für das Canary erstellt werden sollen.
- Lambda-Funktionen und -Ebenen, wenn Sie eine Vorlage verwenden, um das Canary zu erstellen. Diese Ressourcen haben das Präfix `cwsyn-`*`MyCanaryName`*.
- CloudWatch Protokolliert Protokollgruppen mit dem Namen`/aws/lambda/cwsyn-`*`MyCanaryName`*`-`*`randomId`*.

Verwenden von Canary-Vorlagen

Dieser Abschnitt enthält Details zu den jeweiligen Canary-Vorlagen und zu den Aufgaben, für die jede Vorlage am besten geeignet ist. Vorlagen werden für die folgenden Canary-Arten bereitgestellt:

- Heartbeat-Überwachung
- API Canary
- Broken Link Checker
- Visuelle Überwachung
- Canary-Recorder
- GUI Workflow

Wenn Sie einen Blueprint verwenden, um einen Canary zu erstellen, wird beim Ausfüllen der Felder in der CloudWatch Konsole im Skript-Editor-Bereich der Seite der Canary, den Sie gerade erstellen, als Node.js -Skript angezeigt. Sie können Ihr Canary in diesem Bereich auch bearbeiten, um es weiter anzupassen.

Heartbeat-Überwachung

Über Heartbeat-Skripte werden die angegebene URL geladen und ein Screenshot der Seite sowie eine HTTP-Archivdatei (HAR-Datei) gespeichert. Mit ihnen werden auch Protokolle der aufgerufenen URLs gespeichert.

Sie können die HAR-Dateien verwenden, um sich detaillierte Leistungsdaten über die Webseiten anzeigen zu lassen. Sie können die Liste der Web-Anfragen analysieren und Performance-Probleme wie die Ladezeit für ein Element erfassen.

Wenn Ihr Canary die Laufzeitversion `syn-nodejs-puppeteer-3.1` oder höher verwendet, können Sie den Blueprint zur Heartbeat-Überwachung verwenden, um mehrere URLs zu überwachen und den Status, die Dauer, die zugehörigen Screenshots und die Fehlerursache jeder URL in der Schrittzusammenfassung des Canary-Ausführungsberichts anzuzeigen.

API-Canary

Über API Canarys können die grundlegenden Lese- und Schreibfunktionen einer REST-API getestet werden. REST steht für Representational State Transfer und ist eine Reihe von Regeln, die Entwickler bei der Erstellung einer API befolgen. Eine dieser Regeln besagt, dass ein Link zu einer bestimmten URL einen Datenabschnitt zurückgeben soll.

Canaries können mit allen APIs arbeiten und alle Arten von Funktionen testen. Jeder Canary kann mehrere API-Aufrufe ausführen.

In Canaries, die Laufzeitversion `syn-nodejs-2.2` oder höher verwenden, unterstützt der API-Canary-Blueprint mehrstufige Canaries, die Ihre APIs als HTTP-Schritte überwachen. Sie können mehrere APIs in einem einzigen Canary testen. Jeder Schritt ist eine separate Anforderung, die auf eine andere URL zugreifen kann, unterschiedliche Header verwenden und unterschiedliche Regeln dafür verwenden kann, ob Kopfzeilen und Antwortkörper erfasst werden. Indem Sie Kopfzeilen und Antworttext nicht erfassen, können Sie verhindern, dass sensible Daten aufgezeichnet werden.

Jede Anforderung in einem API-Canary besteht aus folgenden Informationen:

- aus dem Endpoint (Endpunkt), d. h. die URL, die Sie anfordern.
- aus der method (Methode), d. h. die Art der Anforderung, die an den Server gesendet wird. REST-APIs unterstützen die Operationen GET (Lesen), POST (Schreiben), PUT (Aktualisieren), PATCH (Aktualisieren) und DELETE (Löschen).

- aus den headers (Headern), die Informationen sowohl für den Client als auch für den Server bereitstellen. Sie werden zur Authentifizierung und Bereitstellung von Informationen über den Body-Inhalt verwendet. Eine Liste der gültigen Header finden Sie unter HTTP-Headers.
- Die data (Daten) (oder der Body), die Informationen enthalten, die an den Server gesendet werden sollen. Diese werden nur für POST-, PUT-, PATCH- oder DELETE-Anforderungen verwendet.

Die API-Canary-Vorlage unterstützt GET- und POST-Methoden. Wenn Sie diese Vorlage verwenden, müssen Sie Header angeben. Sie können z. B. die **`Authorization`** als Key (Schlüssel) angeben und die erforderlichen Autorisierungsdaten als Value (Wert) für diesen Schlüssel angeben.

Wenn Sie eine POST-Anforderung testen, geben Sie auch den zu sendenden Inhalt im Feld Data (Daten) an.

Integration mit API Gateway

Der API-Blueprint ist in Amazon API Gateway integriert. Auf diese Weise können Sie eine API Gateway-API und Staging aus demselben AWS Konto und derselben Region wie der Canary auswählen oder eine Swagger-Vorlage für die konto- und regionsübergreifende API-Überwachung von API Gateway hochladen. Sie können dann die restlichen Details in der Konsole auswählen, um den Canary zu erstellen, anstatt sie von Grund auf neu einzugeben. Weitere Informationen zum API Gateway finden Sie unter Was ist Amazon API Gateway?.

Verwenden einer privaten API

Sie können in Amazon API Gateway einen Canary erstellen, der eine private API verwendet. Weitere Informationen finden Sie unter Private API in Amazon API Gateway erstellen.

Broken Link Checker

Der Broken Link Checker sammelt alle Links innerhalb der URL, die Sie mit `document.getElementsByTagName('a')` testen. Er testet nur bis zu der von Ihnen angegebenen Anzahl von Links, wobei die URL selbst als erster Link gezählt wird. Wenn Sie beispielsweise alle Links auf einer Seite überprüfen möchten, die fünf Links enthält, müssen Sie angeben, dass das Canary sechs Links verfolgen soll.

Canarys für Broken Link Checker, die mit der `syn-nodejs-2.0-beta`-Laufzeit oder höher erstellt wurden, unterstützen die folgenden zusätzlichen Features:

- Stellt einen Bericht bereit, der die überprüften Links, Statuscode, Fehlergrund (falls vorhanden) sowie Screenshots der Quell- und Zielseite enthält.

- Wenn Sie Canary-Ergebnisse anzeigen, können Sie filtern, um nur die fehlerhaften Links anzuzeigen, und dann den Link basierend auf dem Grund des Fehlers beheben.
- Diese Version erfasst für jeden Link kommentierte Quellseiten-Screenshots und hebt den Anker hervor, in dem der Link gefunden wurde. Ausgeblendete Komponenten werden nicht mit Anmerkungen versehen.
- Sie können diese Version so konfigurieren, dass Screenshots von Quell- und Zielseiten, nur Quellseiten oder nur Zielseiten erfasst werden.
- Diese Version behebt ein Problem in der früheren Version, bei dem das Canary-Skript nach dem ersten gebrochenen Link stoppt, selbst wenn mehr Links von der ersten Seite gescrapt werden.

Wenn Sie einen vorhandenen Canary mit `syn-1.0` aktualisieren möchten, um die neue Laufzeit zu verwenden, müssen Sie den Canary löschen und neu erstellen. Durch das Aktualisieren eines vorhandenen Canarys auf die neue Laufzeitumgebung werden diese Features nicht verfügbar gemacht.

Ein Broken Link Checker Canary (Canary für fehlerhafte Links) erkennt die folgenden Arten von Verbindungsfehlern:

- 404 Seite nicht gefunden
- Ungültiger Hostname
- Falsche URL. In der URL fehlt z. B. eine Klammer, sie weist zusätzliche Schrägstriche auf oder es wird das falsche Protokoll verwendet.
- Ungültiger HTTP-Antwortcode
- Der Hostserver gibt leere Antworten ohne Inhalt und ohne Antwortcode zurück.
- Die HTTP-Anforderungen weisen während der Canary-Ausführung ständig Zeitüberschreitungen auf.
- Der Host bricht immer wieder Verbindungen ab, weil er falsch konfiguriert oder überlastet ist.

Blueprint für die visuelle Überwachung

Der visuelle Monitoring-Blueprint enthält Code, um Screenshots, die während eines Canary-Laufs aufgenommen wurden, mit Screenshots zu vergleichen, die während eines Baseline-Canary-Laufs aufgenommen wurden. Wenn die Diskrepanz zwischen den beiden Screenshots einen Schwellenwert überschreitet, schlägt der Canary fehl. Die visuelle Überwachung wird auf Canaries unterstützt,

auf denen syn-puppeteer-node-3.2 und höher ausgeführt wird. Es wird derzeit nicht in Canarys unterstützt, die Python und Selenium ausführen.

Der Blueprint für die visuelle Überwachung enthält die folgende Codezeile im Canary-Standardskript für Blueprints, die die visuelle Überwachung ermöglicht.

```
syntheticsConfiguration.withVisualCompareWithBaseRun(true);
```

Wenn der Canary zum ersten Mal erfolgreich ausgeführt wird, nachdem diese Zeile zum Skript hinzugefügt wurde, verwendet er die während dieser Ausführung erstellten Screenshots als Vergleichsbasis. Nach dem ersten Canary-Run kannst du die CloudWatch Konsole verwenden, um den Canary zu bearbeiten, um einen der folgenden Schritte auszuführen:

- Legen Sie den nächsten Lauf des Canarys als neue Basislinie fest.
- Zeichnen Sie Grenzen auf dem aktuellen Baseline-Screenshot, um Bereiche des Screenshots festzulegen, die bei visuellen Vergleichen ignoriert werden sollen.
- Entfernen Sie einen Screenshot, der für die visuelle Überwachung verwendet wird.

Weitere Informationen zur Verwendung der CloudWatch Konsole zum Bearbeiten eines Canary finden Sie unterEinen Canary bearbeiten oder löschen.

Sie können den Canary-Run, der als Baseline verwendet wird, auch ändern, indem Sie die `lastrun` Parameter `nextrun` oder verwenden oder eine Canary-Run-ID in der UpdateCanaryAPI angeben.

Wenn Sie den Blueprint für die visuelle Überwachung verwenden, geben Sie die URL ein, unter der der Screenshot erstellt werden soll, und geben einen Differenzschwellenwert als Prozentsatz an. Nach dem Baseline-Lauf lösen zukünftige Läufe des Canary, die einen visuellen Unterschied größer als dieser Schwellenwert erkennen, einen Canary-Fehler aus. Nach dem Baseline-Lauf können Sie den Canary auch bearbeiten, um Grenzen auf dem Baseline-Screenshot zu „zeichnen“, die Sie während der visuellen Überwachung ignorieren möchten.

Die visuelle Überwachungsfunktion basiert auf dem ImageMagick Open-Source-Software-Toolkit. Weitere Informationen finden Sie unter ImageMagick.

Canary-Recorder

Mit dem Canary Recorder-Blueprint können Sie den CloudWatch Synthetics Recorder verwenden, um Ihre Klick- und Tippaktionen auf einer Website aufzuzeichnen und automatisch ein Node.js -

Skript zu generieren, mit dem Sie einen Canary erstellen können, der dieselben Schritte ausführt. Der CloudWatch Synthetics Recorder ist eine von Amazon bereitgestellte Google Chrome-Erweiterung.

Credits: Der CloudWatch Synthetics Recorder basiert auf dem Headless-Recorder.

Weitere Informationen finden Sie unter Verwenden des CloudWatch Synthetics Recorders für Google Chrome.

GUI Workflow Builder

Mit der GUI-Workflow-Builder-Vorlage wird überprüft, ob Aktionen auf Ihrer Webseite ausgeführt werden können. Wenn Sie zum Beispiel eine Webseite mit einem Anmeldeformular haben, kann das Canary die Benutzer- und Passwortfelder ausfüllen und das Formular abschicken, um zu überprüfen, ob die Webseite ordnungsgemäß funktioniert.

Wenn Sie eine Vorlage verwenden, um diesen Typ von Canary zu erstellen, geben Sie die Aktionen an, die das Canary auf der Webseite ausführen soll. Folgende Aktionen können Sie verwenden:

- Click (Klicken) – Wählt das von Ihnen angegebene Element aus und simuliert einen Benutzer, der auf das Element klickt oder es auswählt.

  Um das Element in einem Skript „Node.js“ anzugeben, verwenden Sie `[id=]` oder `a[class=]`.

  Um das Element in einem Python-Skript anzugeben, verwenden Sie `xpath //*[@id=]` oder `//*[@class=]`.
- Verify selector (Auswahl überprüfen) – Überprüft, ob das angegebene Element auf der Webseite vorhanden ist. Dieser Test ist nützlich, um zu überprüfen, ob eine vorherige Aktion dazu geführt hat, dass die Seite mit den richtigen Elementen ausgefüllt wird.

  Um das zu verifizierende Element in einem Skript „Node.js“ anzugeben, verwenden Sie `[id=]` oder `a[class=]`.

  Um das zu überprüfende Element in einem Python-Skript anzugeben, verwenden Sie `xpath //*[@id=]` oder `//*[class=]`.
- Verify text (Text überprüfen) – Überprüft, ob die angegebene Zeichenfolge im Zielelement enthalten ist. Dieser Test ist nützlich, um zu überprüfen, ob eine vorherige Aktion dazu geführt hat, dass der richtige Text angezeigt wird.

  Um das Element in einem Node.js-Skript anzugeben, verwenden Sie ein Format wie `div[@id=]//h1`, da diese Aktion die `waitForXPath`-Funktion in Puppeteer verwendet.

Um das Element in einem Python-Skript anzugeben, verwenden Sie das xpath-Format wie `//*[@id=]` oder //*[@class=], da diese Aktion die `implicitly_wait`-Funktion in Selenium verwendet.

- Input text (Eingabetext) – Schreibt den angegebenen Text in das Zielelement.

  Um das zu verifizierende Element in einem Skript „Node.js“ anzugeben, verwenden Sie `[id=]` oder `a[class=]`.

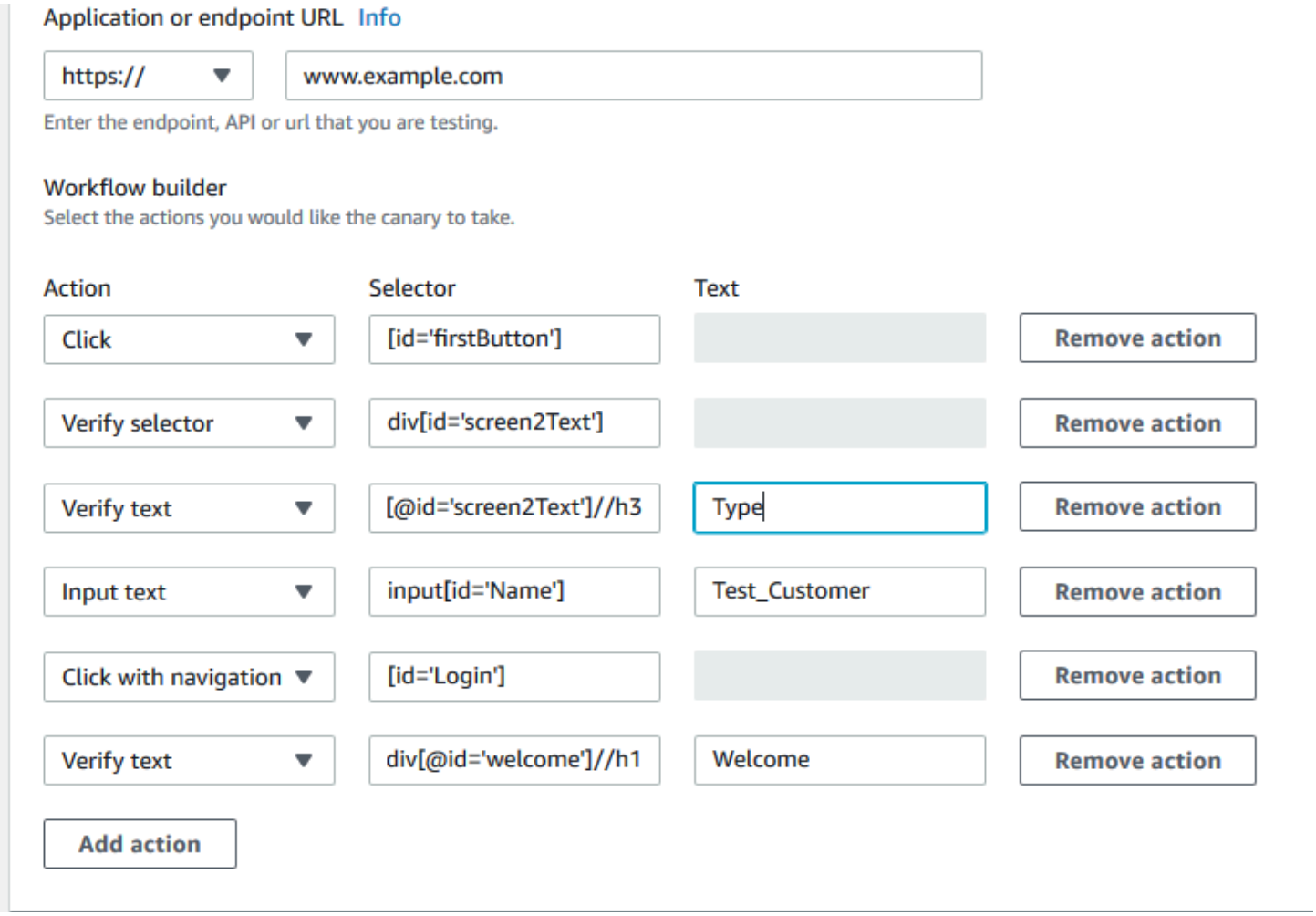
  Um das zu überprüfende Element in einem Python-Skript anzugeben, verwenden Sie `xpath //*[@id=]` oder `//*[@class=]`.

- Click with navigation (Klicken bei Navigation) – Wartet auf das Laden der gesamten Seite, nachdem das angegebene Element ausgewählt wurde. Dies ist besonders nützlich, wenn Sie die Seite neu laden müssen.

  Um das Element in einem Skript „Node.js“ anzugeben, verwenden Sie `[id=]` oder `a[class=]`.

  Um das Element in einem Python-Skript anzugeben, verwenden Sie `xpath //*[@id=]` oder `//*[@class=]`.

Der folgende Blueprint verwendet beispielsweise Node.js. Es wird auf zuerst auf firstButton in der angegebenen URL geklickt und überprüft, ob die erwartete Auswahl mit dem erwarteten Text angezeigt wird, der Name `Test_Customer` wird in das Feld Name eingegeben, es wird auf die Schaltfläche Login (Anmelden) geklickt und dann überprüft, ob die Anmeldung erfolgreich ist, indem auf der nächsten Seite nach dem Text Welcome gesucht wird.



GUI-Workflow-Canaries, die die folgenden Laufzeiten verwenden, bieten auch eine Zusammenfassung der Schritte, die für jeden Canary-Lauf ausgeführt werden. Sie können die Screenshots und die Fehlermeldung, die jedem Schritt zugeordnet sind, verwenden, um die Ursache des Fehlers zu finden.

- `syn-nodejs-2.0` oder höher
- `syn-python-selenium-1.0` oder höher

Verwenden des CloudWatch Synthetics Recorders für Google Chrome

Amazon bietet einen CloudWatch Synthetics Recorder, mit dem Sie Kanarienvögel einfacher erstellen können. Der Recorder ist eine Google-Chrome-Erweiterung.

Der Recorder zeichnet Ihre Klick- und Tippaktionen auf einer Website auf und generiert automatisch ein Node.js Skript, das verwendet werden kann, um einen Canary zu erstellen, der denselben Schritten folgt.

Nachdem Sie mit der Aufnahme begonnen haben, erkennt der CloudWatch Synthetics Recorder Ihre Aktionen im Browser und konvertiert sie in ein Skript. Sie können die Aufzeichnung bei Bedarf anhalten und fortsetzen. Wenn Sie die Aufnahme beenden, erstellt der Rekorder ein Node.js-Skript Ihrer Aktionen, das Sie mit der Schaltfläche In Zwischenablage kopieren einfach kopieren können. Sie können dieses Skript dann verwenden, um einen Kanarienvogel in CloudWatch Synthetics zu erstellen.

Credits: Der CloudWatch Synthetics Recorder basiert auf dem Headless-Recorder.

Installation der CloudWatch Synthetics Recorder-Erweiterung für Google Chrome

Um den CloudWatch Synthetics Recorder zu verwenden, können Sie mit der Erstellung eines Canary beginnen und den Canary Recorder-Blueprint auswählen. Wenn Sie dies tun, obwohl Sie den Rekorder noch nicht heruntergeladen haben, bietet die CloudWatch Synthetics-Konsole einen Link zum Herunterladen.

Alternativ können Sie diese Schritte ausführen, um den Rekorder direkt herunterzuladen und zu installieren.

So installieren Sie den CloudWatch Synthetics Recorder

1. Rufen Sie mit Google Chrome diese Website auf: https://chrome.google.com/webstore/detail/cloudwatch-synthetics-rec /bhdnlmmgiplmbcdmkkdfplenecpegfno
2. Wählen Sie Zu Chrome hinzufügen und dann Erweiterung hinzufügen aus.

Verwenden des CloudWatch Synthetics Recorders für Google Chrome

Wenn Sie mit dem CloudWatch Synthetics Recorder einen Canary erstellen möchten, können Sie in der CloudWatch Konsole Create Canary auswählen und dann Blueprint verwenden, Canary Recorder auswählen. Weitere Informationen finden Sie unter Erstellen eines Canarys.

Alternativ können Sie den Recorder verwenden, um Schritte aufzuzeichnen, ohne sie sofort zum Erstellen eines Canary zu verwenden.

Um den CloudWatch Synthetics Recorder zu verwenden, um Ihre Aktionen auf einer Website aufzuzeichnen

1. Navigieren Sie zu der Seite, die Sie überwachen möchten.
2. Wählen Sie das Chrome-Erweiterungssymbol und dann CloudWatchSynthetics Recorder.
3. Wählen Sie Aufnahme starten.
4. Führen Sie die Schritte aus, die von aufgezeichnet werden sollen. Um die Aufzeichnung anzuhalten, wählen Sie Pause aus.
5. Wenn Sie mit der Aufzeichnung des Workflows fertig sind, wählen Sie Aufzeichnung beenden.
6. Wählen Sie In Zwischenablage kopieren, um das generierte Skript in Ihre Zwischenablage zu kopieren. Wenn Sie von vorne beginnen möchten, wählen Sie Neue Aufnahme.
7. Um einen Canary mit dem kopierten Skript zu erstellen, können Sie das kopierte Skript in den Inline-Editor des Recorder-Blueprints einfügen oder es in einem Amazon-S3-Bucket speichern und von dort importieren.
8. Wenn Sie nicht sofort einen Canary erstellen, können Sie das aufgezeichnete Skript in einer Datei speichern.

Bekannte Einschränkungen des CloudWatch Synthetics Recorders

Der CloudWatch Synthetics Recorder für Google Chrome hat derzeit die folgenden Einschränkungen.

- HTML-Elemente, die keine IDs haben, verwenden CSS-Selektoren. Dies kann Canarys brechen, wenn sich die Webseitenstruktur später ändert. Wir planen, einige Konfigurationsoptionen (wie die Verwendung von Daten-ID) um diese in einer zukünftigen Version des Recorders bereitzustellen.
- Der Recorder unterstützt keine Aktionen wie Doppelklicken oder Kopieren/Einfügen und unterstützt keine Tastenkombinationen wie CMD+0.
- Um zu überprüfen, ob ein Element oder Text auf der Seite vorhanden ist, müssen Benutzer nach der Generierung des Skripts Assertionen hinzufügen. Der Recorder unterstützt die Überprüfung eines Elements nicht, ohne eine Aktion für dieses Element auszuführen. Dies ähnelt den Optionen „Text überprüfen“ oder „Element verifizieren“ im Canary Workflow Builder. Wir planen, einige Behauptungen Unterstützung in einer zukünftigen Version des Recorders hinzuzufügen.
- Der Recorder zeichnet alle Aktionen auf der Registerkarte auf, auf der die Aufnahme initiiert wird. Es werden keine Pop-ups aufgezeichnet (zum Beispiel, um Standortverfolgung zu ermöglichen) oder die Navigation zu verschiedenen Seiten aus Pop-ups.

Synthetics Laufzeitversionen

Wenn Sie ein Canary erstellen oder aktualisieren, wählen Sie eine Synthetics-Laufzeitversion für das Canary aus. Eine Synthetics-Laufzeit ist eine Kombination aus Synthetics-Code, der Ihren Skript-Handler aufruft, und den Lambda-Ebenen gebündelter Abhängigkeiten.

CloudWatch Synthetics unterstützt derzeit Laufzeiten, die Node.js für Skripts und das Puppeteer-Framework verwenden, sowie Laufzeiten, die Python für das Scripting und Selenium Webdriver für das Framework verwenden.

Wir empfehlen, immer die aktuellste Laufzeitversion für Ihre Canarys zu verwenden, um die neuesten Features und Aktualisierungen der Synthetics-Bibliothek nutzen zu können.

Wenn Sie einen Kanarienvogel erstellen, handelt es sich bei einer der erstellten Ebenen um eine Synthetics-Ebene, der vorangestellt ist. `Synthetics` Diese Ebene gehört dem Synthetics-Dienstkonto und enthält den Laufzeitcode.

> **Note**
>
> Wenn Sie ein Canary auf eine neue Version der Synthetics-Laufzeitumgebung aktualisieren, werden auch alle Synthetics-Bibliotheksfunktionen, die Ihr Canary verwendet, automatisch auf die gleiche Version von NodeJS aktualisiert, die die Synthetics-Laufzeitumgebung unterstützt.

Themen

CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime

Synthetics-Laufzeitversionen unterliegen Wartungs- und Sicherheitsupdates. Wenn eine Komponente einer Laufzeitversion nicht mehr unterstützt wird, gilt diese Synthetics-Laufzeitversion als veraltet.

Sie können keine Canaries mit veralteten Laufzeitversionen erstellen. Canaries, die veraltete Laufzeiten verwenden, werden weiterhin ausgeführt. Sie können diese Canaries stoppen, starten und löschen. Sie können ein vorhandenes Canary aktualisieren, das eine veraltete Laufzeitversion

verwendet, indem Sie das Canary so aktualisieren, dass es eine unterstützte Laufzeitversion verwendet.

CloudWatch Synthetics benachrichtigt Sie per E-Mail, wenn Sie Kanarienvögel haben, die Laufzeiten verwenden, die in den nächsten 60 Tagen nicht mehr unterstützt werden sollen. Es wird empfohlen, Ihre Canarys auf eine unterstützte Laufzeitversion zu migrieren, um von den neuen Funktionalitäts-, Sicherheits- und Leistungsverbesserungen zu profitieren, die in neueren Versionen enthalten sind.

Wie aktualisiere ich einen Canary auf eine neue Laufzeitversion?

Sie können die Runtime-Version eines Canary mithilfe der CloudWatch Konsole, AWS CloudFormation des oder des SDK aktualisieren. AWS CLI AWS Wenn du die CloudWatch Konsole verwendest, kannst du bis zu fünf Canaries gleichzeitig aktualisieren, indem du sie auf der Canary-Listenseite auswählst und dann Aktionen, Laufzeit aktualisieren auswählst.

Sie können das Upgrade überprüfen, indem Sie zuerst den Canary mithilfe der CloudWatch Konsole klonen und seine Runtime-Version aktualisieren. Dadurch entsteht ein weiterer Canary, der ein Klon Ihres ursprünglichen Canarys ist. Sobald Sie Ihren Canary mit der neuen Laufzeitversion verifiziert haben, können Sie die Laufzeitversion Ihres ursprünglichen Canary aktualisieren und den Klon-Canary löschen.

Sie können auch mehrere Canarys mit einem Upgrade-Skript aktualisieren. Weitere Informationen finden Sie unter Canary-Laufzeit-Upgradeskript.

Wenn Sie einen Canary aktualisieren und dies fehlschlägt, finden Sie unter Problembehandlung bei fehlgeschlagenem Canary.

Datumsangaben für die Laufzeitabweichung

Laufzeitversion	Datum der Veraltung
`syn-nodejs-puppeteer-6.1`	8. März 2024
`syn-nodejs-puppeteer-6.0`	8. März 2024
`syn-nodejs-puppeteer-5.1`	8. März 2024

Laufzeitversion	Datum der Veraltung
`syn-nodejs-puppeteer-5.0`	8. März 2024
`syn-nodejs-puppeteer-4.0`	8. März 2024
`syn-nodejs-puppeteer-3.9`	8. Januar 2024
`syn-nodejs-puppeteer-3.8`	8. Januar 2024
`syn-python-selenium-2.0`	8. März 2024
`syn-python-selenium-1.3`	8. März 2024
`syn-python-selenium-1.2`	8. März 2024
`syn-python-selenium-1.1`	8. März 2024
`syn-python-selenium-1.0`	8. März 2024
`syn-nodejs-puppeteer-3.7`	8. Januar 2024
`syn-nodejs-puppeteer-3.6`	8. Januar 2024
`syn-nodejs-puppeteer-3.5`	8. Januar 2024
`syn-nodejs-puppeteer-3.4`	13. November 2022

Laufzeitversion	Datum der Veraltung
syn-nodejs-puppeteer-3.3	13. November 2022
syn-nodejs-puppeteer-3.2	13. November 2022
syn-nodejs-puppeteer-3.1	13. November 2022
syn-nodejs-puppeteer-3.0	13. November 2022
syn-nodejs-2.2	28. Mai 2021
syn-nodejs-2.1	28. Mai 2021
syn-nodejs-2.0	28. Mai 2021
syn-nodejs-2.0-beta	8. Februar 2021
syn-1.0	28. Mai 2021

Canary-Laufzeit-Upgradeskript

Verwenden Sie das folgende Skript, um ein Canary-Skript auf eine unterstützte Laufzeitversion zu aktualisieren.

```
const AWS = require('aws-sdk');

// You need to configure your AWS credentials and Region.
//   https://docs.aws.amazon.com/sdk-for-javascript/v3/developer-guide/setting-credentials-node.html
//   https://docs.aws.amazon.com/sdk-for-javascript/v3/developer-guide/setting-region.html

const synthetics = new AWS.Synthetics();
```

```
const DEFAULT_OPTIONS = {
  /**
   * The number of canaries to upgrade during a single run of this script.
   */
  count: 10,
  /**
   * No canaries are upgraded unless force is specified.
   */
  force: false
};

/**
 * The number of milliseconds to sleep between GetCanary calls when
 * verifying that an update succeeded.
 */
const SLEEP_TIME = 5000;

(async () => {
  try {
    const options = getOptions();

    const versions = await getRuntimeVersions();
    const canaries = await getAllCanaries();
    const upgrades = canaries
      .filter(canary => !versions.isLatestVersion(canary.RuntimeVersion))
      .map(canary => {
        return {
          Name: canary.Name,
          FromVersion: canary.RuntimeVersion,
          ToVersion: versions.getLatestVersion(canary.RuntimeVersion)
        };
      });

    if (options.force) {
      const promises = [];

      for (const upgrade of upgrades.slice(0, options.count)) {
        const promise = upgradeCanary(upgrade);
        promises.push(promise);
        // Sleep for 100 milliseconds to avoid throttling.
        await usleep(100);
      }

      const succeeded = [];
```

```
      const failed = [];
      for (let i = 0; i < upgrades.slice(0, options.count).length; i++) {
        const upgrade = upgrades[i];
        const promise = promises[i];
        try {
          await promise;
          console.log(`The update of ${upgrade.Name} succeeded.`);
          succeeded.push(upgrade.Name);
        } catch (e) {
          console.log(`The update of ${upgrade.Name} failed with error: ${e}`);
          failed.push({
            Name: upgrade.Name,
            Reason: e
          });
        }
      }

      if (succeeded.length) {
        console.group('The following canaries were upgraded successfully.');
        for (const name of succeeded) {
          console.log(name);
        }
        console.groupEnd()
      } else {
        console.log('No canaries were upgraded successfully.');
      }

      if (failed.length) {
        console.group('The following canaries were not upgraded successfully.');
        for (const failure of failed) {
          console.log('\x1b[31m', `${failure.Name}: ${failure.Reason}`, '\x1b[0m');
        }
        console.groupEnd();
      }
    } else {
      console.log('Run with --force [--count <count>] to perform the first <count>
 upgrades shown. The default value of <count> is 10.')
      console.table(upgrades);
    }
  } catch (e) {
    console.error(e);
  }
})();
```

```
function getOptions() {
  const force = getFlag('--force', DEFAULT_OPTIONS.force);
  const count = getOption('--count', DEFAULT_OPTIONS.count);
  return { force, count };

  function getFlag(key, defaultValue) {
    return process.argv.includes(key) || defaultValue;
  }
  function getOption(key, defaultValue) {
    const index = process.argv.indexOf(key);
    if (index < 0) {
      return defaultValue;
    }
    const value = process.argv[index + 1];
    if (typeof value === 'undefined' || value.startsWith('-')) {
      throw `The ${key} option requires a value.`;
    }
    return value;
  }
}

function getAllCanaries() {
  return new Promise((resolve, reject) => {
    const canaries = [];

    synthetics.describeCanaries().eachPage((err, data) => {
      if (err) {
        reject(err);
      } else {
        if (data === null) {
          resolve(canaries);
        } else {
          canaries.push(...data.Canaries);
        }
      }
    });
  });
}

function getRuntimeVersions() {
  return new Promise((resolve, reject) => {
    const jsVersions = [];
    const pythonVersions = [];
    synthetics.describeRuntimeVersions().eachPage((err, data) => {
```

```
      if (err) {
        reject(err);
      } else {
        if (data === null) {
          jsVersions.sort((a, b) => a.ReleaseDate - b.ReleaseDate);
          pythonVersions.sort((a, b) => a.ReleaseDate - b.ReleaseDate);
          resolve({
            isLatestVersion(version) {
              const latest = this.getLatestVersion(version);
              return latest === version;
            },
            getLatestVersion(version) {
              if (jsVersions.some(v => v.VersionName === version)) {
                return jsVersions[jsVersions.length - 1].VersionName;
              } else if (pythonVersions.some(v => v.VersionName === version)) {
                return pythonVersions[pythonVersions.length - 1].VersionName;
              } else {
                throw Error(`Unknown version ${version}`);
              }
            }
          });
        } else {
          for (const version of data.RuntimeVersions) {
            if (version.VersionName === 'syn-1.0') {
              jsVersions.push(version);
            } else if (version.VersionName.startsWith('syn-nodejs-2.')) {
              jsVersions.push(version);
            } else if (version.VersionName.startsWith('syn-nodejs-puppeteer-')) {
              jsVersions.push(version);
            } else if (version.VersionName.startsWith('syn-python-selenium-')) {
              pythonVersions.push(version);
            } else {
              throw Error(`Unknown version ${version.VersionName}`);
            }
          }
        }
      }
    });
  });
}

async function upgradeCanary(upgrade) {
  console.log(`Upgrading canary ${upgrade.Name} from ${upgrade.FromVersion} to
 ${upgrade.ToVersion}`);
```

```
  await synthetics.updateCanary({ Name: upgrade.Name, RuntimeVersion:
 upgrade.ToVersion }).promise();
  while (true) {
    await usleep(SLEEP_TIME);
    console.log(`Getting the state of canary ${upgrade.Name}`);
    const response = await synthetics.getCanary({ Name: upgrade.Name }).promise();
    const state = response.Canary.Status.State;
    console.log(`The state of canary ${upgrade.Name} is ${state}`);
    if (state === 'ERROR' || response.Canary.Status.StateReason) {
      throw response.Canary.Status.StateReason;
    }
    if (state !== 'UPDATING') {
      return;
    }
  }
}

function usleep(ms) {
  return new Promise(resolve => setTimeout(resolve, ms));
}
```

Laufzeitversionen mit Node.js und Puppeteer

Die erste Laufzeitversion für Node.js und Puppeteer hieß `syn-1.0`. Spätere Laufzeitversionen haben die Namenskonvention `syn-`*`language`*`-`*`majorversion`*`.`*`minorversion`*. Beginnend mit `syn-nodejs-puppeteer-3.0` ist die Namenskonvention `syn-`*`language`*`-`*`framework`*`-`*`majorversion`*`.`*`minorversion`*

Ein zusätzliches `-beta`-Suffix zeigt an, dass sich die Laufzeitversion derzeit in einer Beta-Vorschauversion befindet.

Laufzeitversionen mit derselben Hauptversionsnummer sind abwärtskompatibel.

> **Important**
>
> Die folgenden CloudWatch Synthetics-Runtime-Versionen werden voraussichtlich am 8. März 2024 veraltet sein.
>
> - `syn-nodejs-puppeteer-6.1`
> - `syn-nodejs-puppeteer-6.0`
> - `syn-nodejs-puppeteer-5.1`

- `syn-nodejs-puppeteer-5.0`
- `syn-nodejs-puppeteer-4.0`

Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Important

WICHTIG: Das enthaltene AWS SDK für die JavaScript v2-Abhängigkeit wird entfernt und aktualisiert, um AWS SDK für Version JavaScript 3 in einer future Runtime-Version zu verwenden. Sobald dies passiert, können Sie Ihre Canary-Code-Referenzen aktualisieren. Alternativ können Sie weiterhin auf das enthaltene AWS SDK für die JavaScript v2-Abhängigkeit verweisen und es verwenden, indem Sie es als Abhängigkeit zu Ihrer Quellcode-ZIP-Datei hinzufügen.

Hinweise für alle Laufzeitversionen

Stellen Sie bei Verwendung der `syn-nodejs-puppeteer-3.0`-Laufzeitversion sicher, dass Ihr Canary-Skript mit Node.js 12.x kompatibel ist. Wenn Sie eine frühere Version einer `syn-nodejs`-Laufzeitversion verwenden, stellen Sie sicher, dass Ihr Skript mit Node.js 10.x kompatibel ist.

Der Lambda-Code in einem Canary ist so konfiguriert, dass er einen maximalen Speicher von 1 GB hat. Für jede Canary-Ausführung tritt nach Ablauf eines konfigurierten Timeoutwerts eine Zeitüberschreitung ein. Wenn kein Timeout-Wert für einen Canary angegeben ist, wird ein Timeout-Wert CloudWatch ausgewählt, der auf der Frequenz des Canary basiert. Wenn Sie einen Timeout-Wert konfigurieren, legen Sie ihn nicht kürzer als 15 Sekunden fest, um Lambda-Kaltstarts und die Zeit zu ermöglichen, die zum Hochfahren der canary-Instrumentierung benötigt wird.

Note

Die folgenden CloudWatch Synthetics-Runtime-Versionen wurden am 8. Januar 2024 als veraltet eingestuft. Dies liegt daran, dass die Lambda Node.js 14-Laufzeit am 4. Dezember 2023 als AWS Lambda veraltet eingestuft wurde.

- `syn-nodejs-puppeteer-3.9`

- `syn-nodejs-puppeteer-3.8`
- `syn-nodejs-puppeteer-3.7`
- `syn-nodejs-puppeteer-3.6`
- `syn-nodejs-puppeteer-3.5`

Die folgenden CloudWatch Synthetics-Runtime-Versionen wurden am 13. November 2022 als veraltet eingestuft. Dies liegt daran, dass die Lambda Node.js 12-Laufzeit am 14. November 2022 als AWS Lambda veraltet eingestuft wurde.

- `syn-nodejs-puppeteer-3.4`
- `syn-nodejs-puppeteer-3.3`
- `syn-nodejs-puppeteer-3.2`
- `syn-nodejs-puppeteer-3.1`
- `syn-nodejs-puppeteer-3.0`

Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

syn-nodejs-puppeteer-7,0

Die `syn-nodejs-puppeteer-7.0` Runtime ist die neueste Runtime-Version für die Lambda-Laufzeit Node.js 18.x. Sie verwendet Node.js und Puppeteer.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 18.x
- Puppeteer-Core-Version 21.9.0
- Chromium-Version 121.0.6167.139

Größe des Codes:

Die Größe des Codes und der Abhängigkeiten, die Sie in diese Runtime packen können, beträgt 80 MB.

Neue Funktionen in syn-nodejs-puppeteer -7.0:

- Aktualisierte Versionen der mitgelieferten Bibliotheken in Puppeteer und Chromium — Die Abhängigkeiten von Puppeteer und Chromium wurden auf neue Versionen aktualisiert.

> **Important**
>
> Die Umstellung von Puppeteer 19.7.0 auf Puppeteer 21.9.0 bringt grundlegende Änderungen in Bezug auf Tests und Filter mit sich. Weitere Informationen finden Sie in den Abschnitten WICHTIGE ÄNDERUNGEN in puppeteer: v20.0.0 und puppeteer-core: v21.0.0.

Es AWS wird ein Upgrade auf SDK v3 empfohlen

Die Lambda-Laufzeit nodejs18.x unterstützt SDK v2 nicht. AWS Wir empfehlen dringend, auf SDK v3 zu migrieren. AWS

syn-nodejs-puppeteer-6.2

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 18.x
- Puppeteer-Core-Version 19.7.0
- Chromium-Version 111.0.5563.146

Neue Funktionen in -6.2: syn-nodejs-puppeteer

- Aktualisierte Versionen der gebündelten Bibliotheken in Chromium
- Ephemere Speicherüberwachung — Diese Runtime fügt die kurzlebige Speicherüberwachung in Kundenkonten hinzu.
- Fehlerkorrekturen

syn-nodejs-puppeteer-5.2

Die `syn-nodejs-puppeteer-5.2` Runtime ist die neueste Runtime-Version für die Lambda-Laufzeit Node.js 16.x. Sie verwendet Node.js und Puppeteer.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 16.x
- Puppeteer-Core-Version 19.7.0
- Chromium-Version 111.0.5563.146

Neue Funktionen in syn-nodejs-puppeteer -5.2:

- Aktualisierte Versionen der gebündelten Bibliotheken in Chromium
- Fehlerkorrekturen

syn-nodejs-puppeteer-6.1

> **Important**
>
> Diese Runtime-Version wird voraussichtlich am 8. März 2024 als veraltet eingestuft. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 18.x
- Puppeteer-Core-Version 19.7.0
- Chromium-Version 111.0.5563.146

Neue Funktionen in syn-nodejs-puppeteer -6.1:

- Verbesserungen der Stabilität – Automatische Wiederholungslogik für den Umgang mit intermittierenden Puppeteer-Startfehlern hinzugefügt.
- Abhängigkeits-Upgrades – Aktualisiert einige Abhängigkeitspakete von Drittanbietern.
- Canarys ohne Amazon-S3-Berechtigungen – Fehlerkorrekturen, sodass Canarys, die keine Amazon-S3-Berechtigungen haben, weiterhin ausgeführt werden können. Diese Canarys ohne Amazon-S3-Berechtigungen können keine Screenshots oder andere Artefakte auf Amazon S3 hochladen. Weitere Informationen zu den Berechtigungen für Canarys finden Sie unter Erforderliche Rollen und Berechtigungen für Canarys.

> **Important**
>
> WICHTIG: Das enthaltene AWS SDK für die JavaScript v2-Abhängigkeit wird entfernt und aktualisiert, um AWS SDK für Version JavaScript 3 in einer future Runtime-Version zu verwenden. Sobald dies passiert, können Sie Ihre Canary-Code-Referenzen aktualisieren. Alternativ können Sie weiterhin auf das enthaltene AWS SDK für die JavaScript v2-Abhängigkeit verweisen und es verwenden, indem Sie es als Abhängigkeit zu Ihrer Quellcode-ZIP-Datei hinzufügen.

syn-nodejs-puppeteer-6,0

> **Important**
>
> Diese Runtime-Version wird voraussichtlich am 8. März 2024 als veraltet eingestuft. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 18.x
- Puppeteer-Core-Version 19.7.0
- Chromium-Version 111.0.5563.146

Neue Funktionen in syn-nodejs-puppeteer -6.0:

- Abhängigkeitsupgrade – Die Abhängigkeit Node.js wurde auf Version 18.x verbesser.
- Unterstützung für den Abfang-Modus – Puppeteer unterstützt nun den kooperativen Abfang-Modus in der Laufzeitbibliothek von Synthetics Canary.
- Änderung des Ablaufverfolgungsverhaltens – Das standardmäßige Ablaufverfolgungsverhalten wurde dahingehend geändert, dass nur noch Fetch- und Xhr-Anfragen verfolgt werden und keine Ressourcenanfragen mehr. Sie können die Ablaufverfolgung von Ressourcenanfragen aktivieren, indem Sie die Option `traceResourceRequests` konfigurieren.
- Die Metrik für die Dauer wurde verfeinert — Die `Duration` Metrik schließt jetzt die Betriebszeit aus, die der Canary zum Hochladen von Artefakten, zum Erstellen von Screenshots und zum

Generieren CloudWatch von Metriken verwendet. `Duration`Metrikwerte werden gemeldet CloudWatch, und Sie können sie auch in der Synthetics-Konsole sehen.

- Fehlerbehebung: – Bereinigen des Core-Dumps, der generiert wurde, wenn Chromium während eines Canary-Laufs abstürzt.

> **Important**
>
> WICHTIG: Das enthaltene AWS SDK für die JavaScript v2-Abhängigkeit wird entfernt und aktualisiert, um AWS SDK für Version JavaScript 3 in einer future Runtime-Version zu verwenden. Sobald dies passiert, können Sie Ihre Canary-Code-Referenzen aktualisieren. Alternativ können Sie weiterhin auf das enthaltene AWS SDK für die JavaScript v2-Abhängigkeit verweisen und es verwenden, indem Sie es als Abhängigkeit zu Ihrer Quellcode-ZIP-Datei hinzufügen.

syn-nodejs-puppeteer-5.1

> **Important**
>
> Diese Runtime-Version wird voraussichtlich am 8. März 2024 als veraltet eingestuft. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 16.x
- Puppeteer-Core-Version 19.7.0
- Chromium-Version 111.0.5563.146

Fehlerkorrekturen in syn-nodejs-puppeteer -5.1:

- Fehlerbehebung – Diese Laufzeit behebt einen Fehler in `syn-nodejs-puppeteer-5.0`, bei dem in den von den Canarys erstellten HAR-Dateien Anforderungsheader fehlten.

syn-nodejs-puppeteer-5.0

> **Important**
>
> Diese Runtime-Version wird voraussichtlich am 8. März 2024 als veraltet eingestuft. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 16.x
- Puppeteer-Core-Version 19.7.0
- Chromium-Version 111.0.5563.146

Neue Funktionen in syn-nodejs-puppeteer -5.0:

- Abhängigkeitsupgrade – Die Puppeteer-Core-Version wurde auf 19.7.0 aktualisiert. Die Chromium-Version wurde auf 111.0.5563.146 aktualisiert.

> **Important**
>
> Die neue Puppeteer-Core-Version ist nicht vollständig mit früheren Versionen von Puppeteer rückwärtskompatibel. Einige der Änderungen in dieser Version können dazu führen, dass bestehende Canarys, die veraltete Puppeteer-Funktionen verwenden, fehlschlagen. Weitere Informationen finden Sie in den Änderungsprotokollen für die Puppeteer-Core-Versionen 19.7.0 bis 6.0 unter Puppeteer-Änderungsprotokolle.

syn-nodejs-puppeteer-4.0

> **Important**
>
> Diese Runtime-Version wird voraussichtlich am 8. März 2024 als veraltet eingestuft. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 16.x
- Puppeteer-Core-Version 5.5.0
- Chromium-Version 92.0.4512

Neue Funktionen in syn-nodejs-puppeteer -4.0:

- Abhängigkeitsupgrade – Die Abhängigkeit Node.js wurde auf Version 16.x aktualisiert.

Veraltete Laufzeiten für Node.js und Puppeteer

Die folgenden Laufzeiten für Node.js und Puppeteer sind veraltet.

syn-nodejs-puppeteer-3.9

> **Important**
>
> Diese Runtime-Version wurde am 8. Januar 2024 als veraltet eingestuft. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 14.x
- Puppeteer-Core-Version 5.5.0
- Chromium-Version 92.0.4512

Neue Funktionen in syn-nodejs-puppeteer -3.9:

- Abhängigkeitsupgrades – Aktualisiert einige Abhängigkeitspakete von Drittanbietern.

syn-nodejs-puppeteer-3.8

> **Important**
>
> Diese Runtime-Version wurde am 8. Januar 2024 als veraltet eingestuft. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 14.x
- Puppeteer-Core-Version 5.5.0
- Chromium-Version 92.0.4512

Neue Funktionen in syn-nodejs-puppeteer -3.8:

- Profilbereinigung – Chromium-Profile werden jetzt nach jeder Canary-Ausführung bereinigt.

Fehlerkorrekturen in syn-nodejs-puppeteer -3.8:

- Bugfixes – Bisher funktionierte die visuelle Überwachung von Canary manchmal nach einem Lauf ohne Screenshots nicht mehr richtig. Dieses Problem wurde behoben.

syn-nodejs-puppeteer-3.7

> **Important**
>
> Diese Runtime-Version wurde am 8. Januar 2024 als veraltet eingestuft. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 14.x
- Puppeteer-Core-Version 5.5.0

- Chromium-Version 92.0.4512

Neue Funktionen in syn-nodejs-puppeteer -3.7:

- Verbessertes Protokoll – Der Canary lädt Protokolle auf Amazon S3 hoch, auch wenn die Zeit überschritten wird oder ein Absturz auftritt.
- Lambda-Ebenengröße reduziert – Die Größe der für Canarys verwendeten Lambda-Ebene wird um 34 % reduziert.

Fehlerkorrekturen in syn-nodejs-puppeteer -3.7:

- Fehlerbehebungen – Japanisch, Vereinfachtes Chinesisch und Traditionelles Chinesisch werden ordnungsgemäß wiedergegeben.

syn-nodejs-puppeteer-3.6

> **Important**
>
> Diese Runtime-Version wurde am 8. Januar 2024 als veraltet eingestuft. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 14.x
- Puppeteer-Core-Version 5.5.0
- Chromium-Version 92.0.4512

Neue Funktionen in syn-nodejs-puppeteer -3.6:

- Präzisere Zeitstempel: Start- und Endzeit von Canary-Ausführungen sind jetzt auf die Millisekunde genau.

syn-nodejs-puppeteer-3.5

> **Important**
>
> Diese Runtime-Version wurde am 8. Januar 2024 als veraltet eingestuft. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 14.x
- Puppeteer-Core-Version 5.5.0
- Chromium-Version 92.0.4512

Neue Funktionen in syn-nodejs-puppeteer -3.5:

- Aktualisierte Abhängigkeiten – Die einzigen neuen Features in dieser Laufzeit sind die aktualisierten Abhängigkeiten.

syn-nodejs-puppeteer-3.4

> **Important**
>
> Diese Laufzeitversion ist seit dem 13. November 2022 veraltet. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 12.x
- Puppeteer-Core-Version 5.5.0
- Chrom-Version 88.0.4298.0

Neue Funktionen in -3.4: syn-nodejs-puppeteer

- Benutzerdefinierte Handler-Funktion – Sie können jetzt eine benutzerdefinierte Handler-Funktion für Ihre Canary-Skripte verwenden. Bei früheren Laufzeiten musste der Skript-Eintrittspunkt `.handler` enthalten.

  Außerdem können Sie Canary-Skripte in einem beliebigen Ordner ablegen und den Ordnernamen als Teil des Handlers übergeben. Beispielsweise kann `MyFolder/MyScriptFile.functionname` als Eintrittspunkt verwendet werden.
- Umfassendere Informationen zu HAR Dateien – Sie können jetzt ungültige, ausstehende und unvollständige Anfragen in den HAR-Dateien sehen, die von Canarys erstellt werden.

syn-nodejs-puppeteer-3.3

> **Important**
>
> Diese Laufzeitversion ist seit dem 13. November 2022 veraltet. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 12.x
- Puppeteer-Core-Version 5.5.0
- Chrom-Version 88.0.4298.0

Neue Funktionen in -3.3: syn-nodejs-puppeteer

- Weitere Optionen für die Artefaktverschlüsselung — Für Kanarienvögel, die diese Runtime oder eine spätere Version verwenden, können Sie wählen, ob Sie einen vom AWS KMS Kunden AWS verwalteten Schlüssel oder einen von Amazon S3 verwalteten Schlüssel verwenden möchten, anstatt einen verwalteten Schlüssel zur Verschlüsselung von Artefakten zu verwenden, die der Canary in Amazon S3 speichert. Weitere Informationen finden Sie unter Verschlüsseln von Canary-Artefakten.

syn-nodejs-puppeteer-3.2

> **Important**
>
> Diese Laufzeitversion ist seit dem 13. November 2022 veraltet. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 12.x
- Puppeteer-Core-Version 5.5.0
- Chrom-Version 88.0.4298.0

Neue Funktionen in -3.2: syn-nodejs-puppeteer

- Visuelle Überwachung mit Screenshots – Canarys, die diese Laufzeit oder höher verwenden, können einen während eines Laufs aufgenommenen Screenshot mit einer Baseline-Version desselben Screenshots vergleichen. Wenn sich die Screenshots stärker als ein festgelegter Prozentschwellenwert unterscheiden, schlägt der Canary fehl. Weitere Informationen finden Sie unter Visuelle Überwachung oder Blueprint für die visuelle Überwachung.
- Neue Funktionen bezüglich sensibler Daten – Sie können verhindern, dass sensible Daten in Canary-Protokollen und -Berichten erscheinen. Weitere Informationen finden Sie unter SyntheticsLogHelper Klasse.
- Veraltete Funktion Die `RequestResponseLogHelper`-Klasse ist zugunsten anderer neuer Konfigurationsoptionen veraltet. Weitere Informationen finden Sie unter RequestResponseLogHelper Klasse.

syn-nodejs-puppeteer-3.1

> **Important**
>
> Diese Laufzeitversion ist seit dem 13. November 2022 veraltet. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 12.x
- Puppeteer-Core-Version 5.5.0
- Chrom-Version 88.0.4298.0

Neue Funktionen in -3.1: syn-nodejs-puppeteer

- Möglichkeit, CloudWatch Metriken zu konfigurieren — Mit dieser Runtime können Sie die Metriken deaktivieren, die Sie nicht benötigen. Andernfalls veröffentlichen Canaries verschiedene CloudWatch Metriken für jeden Canary-Run.
- Screenshot-Verknüpfung – Sie können einen Screenshot mit einem Canary-Schritt verknüpfen, nachdem der Schritt abgeschlossen ist. Dazu erstellen Sie den Screenshot mit der Methode takeScreenshot und verwenden den Namen des Schritts, dem Sie den Screenshot zuordnen möchten. Sie können beispielsweise einen Schritt ausführen, eine Wartezeit hinzufügen und dann den Screenshot erstellen.
- Der Heartbeat-Monitor-Blueprint kann mehrere URLs überwachen — Sie können den Heartbeat-Monitoring-Blueprint in der CloudWatch Konsole verwenden, um mehrere URLs zu überwachen und den Status, die Dauer, die zugehörigen Screenshots und die Fehlerursache für jede URL in der Schrittzusammenfassung des Canary-Run-Berichts zu sehen.

syn-nodejs-puppeteer-3,0

> **Important**
>
> Diese Laufzeitversion ist seit dem 13. November 2022 veraltet. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 12.x
- Puppeteer-Core-Version 5.5.0
- Chrom-Version 88.0.4298.0

Neue Funktionen in -3.0: syn-nodejs-puppeteer

- Aktualisierte Abhängigkeiten – Diese Version verwendet Puppeteer Version 5.5.0, Node.js 12.x und Chromium 88.0.4298.0.
- Regionsübergreifender Bucket – Sie können jetzt einen S3 Bucket in einer anderen Region als den Bucket angeben, in dem Ihr Canary seine Protokolldateien, Screenshots und HAR-Dateien speichert.
- Neue Funktionen verfügbar – Diese Version fügt Bibliotheksfunktionen hinzu, um den Canary-Namen und die Synthetics-Laufzeitversion abzurufen.

  Weitere Informationen finden Sie unter Synthetics-Klasse.

syn-nodejs-2.2

Dieser Abschnitt enthält Informationen zur `syn-nodejs-2.2`-Laufzeitversion.

> **Important**
>
> Diese Laufzeitversion wurde am 28. Mai 2021 veraltet. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 10.x
- Puppeteer-Core-Version 3.3.0
- Chrom-Version 83.0.4103.0

Neue Features in syn-nodejs-2.2:

- Überwachen Sie Ihre Canarys als HTTP-Schritte – Sie können jetzt mehrere APIs in einem einzigen Canary testen. Jede API wird als separater HTTP-Schritt getestet, und CloudWatch Synthetics überwacht den Status jedes Schritts anhand von Schrittmetriken und dem CloudWatch Synthetics-Schrittbericht. CloudWatch Synthetics erstellt `SuccessPercent` `Duration` Metriken für jeden HTTP-Schritt.

  Diese Funktionalität wird durch die Funktion executeHttpStep(stepName, RequestOptions, Callback, StepConfig) implementiert. Weitere Informationen finden Sie unter executeHttpStep(stepName, RequestOptions, [Rückruf], [StepConfig]).

Der API-Canary-Blueprint wird aktualisiert, um dieses neue Feature zu verwenden.

- HTTP-Anforderungsberichte – Sie können jetzt detaillierte HTTP-Anforderungsberichte anzeigen, die Details wie Anforderungs-/Antwort-Header, Antworttext, Statuscode, Fehler- und Leistungstimings, TCP-Verbindungszeit, TLS-Handshake-Zeit, erste Byte-Zeit und Inhaltsübertragungszeit erfassen. Alle HTTP-Anfragen, die das HTTP/HTTPS-Modul unter der Haube verwenden, werden hier erfasst. Header und Antworttext werden nicht standardmäßig erfasst, können aber durch Festlegen von Konfigurationsoptionen aktiviert werden.
- Globale Konfiguration und Konfiguration auf schrittweiser Ebene — Sie können CloudWatch Synthetics-Konfigurationen auf globaler Ebene festlegen, die auf alle Stufen der Kanaren angewendet werden. Sie können diese Konfigurationen auch auf Schrittebene überschreiben, indem Sie Konfigurationsschlüssel-Wert-Paare übergeben, um bestimmte Optionen zu aktivieren oder zu deaktivieren.

  Weitere Informationen finden Sie unter SyntheticsConfiguration Klasse.
- Konfiguration bei Schrittfehler fortsetzen – Sie können die Canary-Ausführung fortsetzen, wenn ein Schritt fehlschlägt. Für die `executeHttpStep`-Funktion ist diese standardmäßig aktiviert. Sie können diese Option einmal auf globaler Ebene festlegen oder pro Schritt unterschiedlich festlegen.

syn-nodejs-2.1

> **Important**
>
> Diese Laufzeitversion wurde am 28. Mai 2021 veraltet. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 10.x
- Puppeteer-Core-Version 3.3.0
- Chrom-Version 83.0.4103.0

Neue Features in syn-nodejs-2.1:

- Konfigurierbares Screenshot-Verhalten – Bietet die Möglichkeit, die Aufnahme von Screenshots durch UI-Canarys auszuschalten. In Canarys, die frühere Versionen der Laufzeitumgebungen verwenden, erfassen UI-Canarys immer Screenshots vor und nach jedem Schritt. Bei `syn-nodejs-2.1` ist dies konfigurierbar. Wenn Sie Screenshots deaktivieren, können Sie Ihre Amazon-S3-Speicherkosten senken und Sie dabei unterstützen, die HIPAA-Vorschriften einzuhalten. Weitere Informationen finden Sie unter SyntheticsConfiguration Klasse.
- Anpassen der Startparameter von Google Chrome – Sie können nun die Argumente konfigurieren, die verwendet werden, wenn ein Canary ein Google-Chrome-Browserfenster startet. Weitere Informationen finden Sie unter Start (Optionen).

Bei Verwendung von syn-nodejs-2.0 oder höher kann es im Vergleich zu früheren Versionen der Canary-Laufzeit zu einer geringfügigen Verlängerung der Canary-Dauer kommen.

syn-nodejs-2.0

> **Important**
>
> Diese Laufzeitversion wurde am 28. Mai 2021 veraltet. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 10.x
- Puppeteer-Core-Version 3.3.0
- Chrom-Version 83.0.4103.0

Neue Features in syn-nodejs-2.0:

- Aktualisierte Abhängigkeiten – Diese Laufzeitversion verwendet Puppeteer-Core-Version 3.3.0 und Chromium-Version 83.0.4103.0
- Support für aktives X-Ray-Tracing. Wenn auf einem Canary Tracing aktiviert ist, werden X-Ray-Traces für alle vom Canary getätigten Aufrufe gesendet, die den Browser, das AWS SDK oder HTTP- oder HTTPS-Module verwenden. Canarys mit aktivierter Ablaufverfolgung werden in der X-Ray Trace Map angezeigt, selbst wenn sie keine Anforderungen an andere Services oder Anwendungen senden, für die die Ablaufverfolgung aktiviert ist. Weitere Informationen finden Sie unter Canary- und X-Ray-Ablaufverfolgung.

- Synthetics-Berichterstattung — Für jeden Canary-Lauf erstellt CloudWatch Synthetics einen Bericht mit dem Namen `SyntheticsReport-PASSED.json` oder`SyntheticsReport-FAILED.json`, der Daten wie Startzeit, Endzeit, Status und Fehler aufzeichnet. Es zeichnet auch den Status PASSED/FEILED jedes Schritts des Canary-Skripts sowie für jeden Schritt erfasste Fehler und Screenshots auf.
- Bericht zur Überprüfung für fehlerhafte Links – Die neue Version des defekten Link-Prüfungsprogramms, die in dieser Laufzeit enthalten ist, erstellt einen Bericht, der die überprüften Links, Statuscode, Fehlergrund (falls vorhanden) sowie Screenshots der Quell- und Zielseite enthält.
- Neue CloudWatch Metriken — Synthetics veröffentlicht Metriken mit den Namen `2xx` `4xx5xx`,, und `RequestFailed` im `CloudWatchSynthetics` Namespace. Diese Metriken zeigen die Anzahl der 200s, 400s, 500s und Anforderungsfehler in den Canary-Abläufen an. Mit dieser Laufzeitversion werden diese Metriken nur für UI-Canarys gemeldet und nicht für API-Canarys gemeldet. Sie werden auch für API-Canaries ab Laufzeitversion `syn-nodejs-puppeteeer-2.2` gemeldet.
- Sortierbare HAR-Dateien – Sie können Ihre HAR-Dateien jetzt nach Statuscode, Anforderungsgröße und Dauer sortieren.
- Zeitstempel für CloudWatch Metriken — Metriken werden jetzt auf der Grundlage der Lambda-Aufrufzeit und nicht auf der Grundlage der Canary-Run-Endzeit gemeldet.

Fehlerbehebungen in syn-nodejs-2.0:

- Problem behoben, dass Fehler beim Hochladen von Canary-Artefakten nicht gemeldet wurden. Solche Fehler werden nun als Ausführungsfehler aufgetaucht.
- Es wurde das Problem behoben, dass umgeleitete Anfragen (3xx) fälschlicherweise als Fehler protokolliert wurden.
- Es wurde das Problem behoben, dass Screenshots beginnend mit 0 durchnummeriert wurden. Sie sollten jetzt mit 1 beginnen.
- Das Problem, dass Screenshots für chinesische und japanische Schriftarten verstümmelt wurden, wurde behoben.

Bei Verwendung von syn-nodejs-2.0 oder höher kann es im Vergleich zu früheren Versionen der Canary-Laufzeit zu einer geringfügigen Verlängerung der Canary-Dauer kommen.

syn-nodejs-2.0-beta

> **Important**
>
> Diese Laufzeitversion wurde am 8. Februar 2021 veraltet. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 10.x
- Puppeteer-Core-Version 3.3.0
- Chrom-Version 83.0.4103.0

Neue Features in syn-nodejs-2.0-beta:

- Aktualisierte Abhängigkeiten – Diese Laufzeitversion verwendet Puppeteer-Core-Version 3.3.0 und Chromium-Version 83.0.4103.0
- Synthetics-Berichterstattung — Für jeden Canary-Lauf erstellt CloudWatch Synthetics einen Bericht mit dem Namen `SyntheticsReport-PASSED.json` oder`SyntheticsReport-FAILED.json`, der Daten wie Startzeit, Endzeit, Status und Fehler aufzeichnet. Es zeichnet auch den Status PASSED/FEILED jedes Schritts des Canary-Skripts sowie für jeden Schritt erfasste Fehler und Screenshots auf.
- Bericht zur Überprüfung für fehlerhafte Links – Die neue Version des defekten Link-Prüfungsprogramms, die in dieser Laufzeit enthalten ist, erstellt einen Bericht, der die überprüften Links, Statuscode, Fehlergrund (falls vorhanden) sowie Screenshots der Quell- und Zielseite enthält.
- Neue CloudWatch Metriken — Synthetics veröffentlicht Metriken mit den Namen `2xx` `4xx`5xx,, und `RequestFailed` im `CloudWatchSynthetics` Namespace. Diese Metriken zeigen die Anzahl der 200s, 400s, 500s und Anforderungsfehler in den Canary-Abläufen an. Diese Metriken werden nur für UI-Canary gemeldet und nicht für API-Canary gemeldet.
- Sortierbare HAR-Dateien – Sie können Ihre HAR-Dateien jetzt nach Statuscode, Anforderungsgröße und Dauer sortieren.
- Zeitstempel für CloudWatch Metriken — Metriken werden jetzt auf der Grundlage der Lambda-Aufrufzeit und nicht auf der Grundlage der Canary-Run-Endzeit gemeldet.

Fehlerbehebungen in syn-nodejs-2.0-beta:

- Problem behoben, dass Fehler beim Hochladen von Canary-Artefakten nicht gemeldet wurden. Solche Fehler werden nun als Ausführungsfehler aufgetaucht.
- Es wurde das Problem behoben, dass umgeleitete Anfragen (3xx) fälschlicherweise als Fehler protokolliert wurden.
- Es wurde das Problem behoben, dass Screenshots beginnend mit 0 durchnummeriert wurden. Sie sollten jetzt mit 1 beginnen.
- Das Problem, dass Screenshots für chinesische und japanische Schriftarten verstümmelt wurden, wurde behoben.

syn-1.0

> **Important**
>
> Diese Laufzeitversion ist voraussichtlich am 28. Mai 2021 veraltet. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Die erste Synthetics-Laufzeitversion ist `syn-1.0`.

Wichtige Abhängigkeiten:

- Lambda-Laufzeit Node.js 10.x
- Puppeteer-Core-Version 1.14.0
- Die Chromium-Version, die Puppeteer-Core 1.14.0 entspricht

Laufzeitversionen mit Python und Selenium Webdriver

Die folgenden Abschnitte enthalten Informationen zu den CloudWatch Synthetics-Laufzeitversionen für Python und Selenium Webdriver. Selenium ist ein Open-Source-Browser-Automatisierungs-Tool. Weitere Informationen zu Selenium finden Sie unter www.selenium.dev/.

Die Namenskonvention für diese Laufzeitversionen lautet `syn-`*`language`*`-`*`framework`*`-`*`majorversion`*`.`*`minorversion`*.

Important

Die folgenden CloudWatch Synthetics-Runtime-Versionen werden voraussichtlich am 8. März 2024 veraltet sein.

- `syn-python-selenium-2.0`
- `syn-python-selenium-1.3`
- `syn-python-selenium-1.2`
- `syn-python-selenium-1.1`
- `syn-python-selenium-1.0`

Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

syn-python-selenium-3.0

Version 3.0 ist die neueste CloudWatch Synthetics-Runtime für Python und Selenium.

Wichtige Abhängigkeiten:

- Python 3.8
- Selenium 4.15.1
- Chrom-Version 121.0.6167.139

Neue Funktionen in -3.0: syn-python-selenium

- Aktualisierte Versionen der gebündelten Bibliotheken in Chromium — Die Chromium-Abhängigkeit wurde auf eine neue Version aktualisiert.

syn-python-selenium-2.1

Wichtige Abhängigkeiten:

- Python 3.8
- Selen 4.15.1

- Chromium-Version 111.0.5563.146

Neue Funktionen in -2.1: syn-python-selenium

- Aktualisierte Versionen der gebündelten Bibliotheken in Chromium — Die Chromium - und Selenium-Abhängigkeiten wurden auf neue Versionen aktualisiert.

syn-python-selenium-2.0

> **Important**
>
> Diese Runtime-Version wird voraussichtlich am 8. März 2024 als veraltet eingestuft. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Python 3.8
- Selenium 4.10.0
- Chromium-Version 111.0.5563.146

Neue Funktionen in syn-python-selenium -2.0:

- Aktualisierte Abhängigkeiten – Die Chromium- und Selenium-Abhängigkeiten wurden auf neue Versionen aktualisiert.

Fehlerkorrekturen in syn-python-selenium -2.0:

- Zeitstempel hinzugefügt – Ein Zeitstempel wurde zu Canary-Protokollen hinzugefügt.
- Wiederverwendung von Sitzungen – Es wurde ein Fehler behoben, der verhindert, dass Canarys die Sitzung ihres vorherigen Canary-Laufs wiederverwenden.

syn-python-selenium-1.3

> **Important**
>
> Diese Runtime-Version wird voraussichtlich am 8. März 2024 als veraltet eingestuft. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Python 3.8
- Selenium 3.141.0
- Chromium-Version 92.0.4512.0

Neue Funktionen in syn-python-selenium -1.3:

- Präzisere Zeitstempel: Start- und Endzeit von Canary-Ausführungen sind jetzt auf die Millisekunde genau.

syn-python-selenium-1.2

> **Important**
>
> Diese Runtime-Version wird voraussichtlich am 8. März 2024 als veraltet eingestuft. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Python 3.8
- Selenium 3.141.0
- Chromium-Version 92.0.4512.0

- Aktualisierte Abhängigkeiten – Die einzigen neuen Funktionen in dieser Laufzeit sind die aktualisierten Abhängigkeiten.

syn-python-selenium-1.1

> **Important**
>
> Diese Runtime-Version wird voraussichtlich am 8. März 2024 als veraltet eingestuft. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Python 3.8
- Selenium 3.141.0
- Chrom-Version 83.0.4103.0

Features:

- Benutzerdefinierte Handler-Funktion – Sie können jetzt eine benutzerdefinierte Handler-Funktion für Ihre Canary-Skripte verwenden. Bei früheren Laufzeiten musste der Skript-Eintrittspunkt `.handler` enthalten.

  Außerdem können Sie Canary-Skripte in einem beliebigen Ordner ablegen und den Ordnernamen als Teil des Handlers übergeben. Beispielsweise kann `MyFolder/MyScriptFile.functionname` als Eintrittspunkt verwendet werden.
- Konfigurationsoptionen zum Hinzufügen von Metriken und Schrittfehlerkonfigurationen – Diese Optionen waren bereits in Laufzeiten für Node.js-Canarys verfügbar. Weitere Informationen finden Sie unter SyntheticsConfiguration Klasse.
- Benutzerdefinierte Argumente in Chrome – Sie können jetzt einen Browser im Inkognito-Modus öffnen oder die Proxy-Server-Konfiguration übergeben. Weitere Informationen finden Sie unter Chrome().
- Regionsübergreifende Artefakt-Bucketse – Ein Canary kann seine Artefakte in einem Amazon-S3-Bucket in einer anderen Region speichern.
- Fehlerbehebungen, einschließlich einer Fehlerbehebung für das **`index.py`**-Problem – Bei früheren Laufzeiten führte eine Canary-Datei mit dem Namen `index.py` zu Ausnahmen, weil dies einen Konflikt mit dem Namen der Bibliotheksdatei verursachte. Dieses Problem wurde behoben.

syn-python-selenium-1.0

> **Important**
>
> Diese Runtime-Version wird voraussichtlich am 8. März 2024 als veraltet eingestuft. Weitere Informationen finden Sie unter CloudWatch Richtlinie zur Unterstützung von Synthetics Runtime.

Wichtige Abhängigkeiten:

- Python 3.8
- Selenium 3.141.0
- Chrom-Version 83.0.4103.0

Features:

- Selenium-Unterstützung – Sie können Canary-Skripte mit dem Selenium-Test-Framework schreiben. Sie können Ihre Selenium-Skripte mit minimalen Änderungen von woanders in CloudWatch Synthetics importieren, und sie funktionieren mit AWS Diensten.

Ein Canary-Skript schreiben

In den folgenden Abschnitten wird erklärt, wie man ein Canary-Skript schreibt und wie man ein Canary-Skript in andere AWS Dienste und mit externen Abhängigkeiten und Bibliotheken integriert.

Themen

- Ein Node.js-Canary-Skript schreiben
- Ein Python-Canary-Skript schreiben
- Ändern eines vorhandenen Selenium-Skripts zur Verwendung eines Synthetics-Canarys
- Änderung eines vorhandenen Puppeteer Synthetics-Skripts zur Authentifizierung von nicht standardmäßigen Zertifikaten

Ein Node.js-Canary-Skript schreiben

Themen

- Einen CloudWatch Synthetics-Kanarienvogel von Grund auf neu erstellen
- Verpacken Sie Ihre kanarischen Dateien von Node.js
- Ändern eines vorhandenen Puppeteer-Skripts zur Verwendung als Synthetics-Canary
- Umgebungsvariablen
- Integrieren Sie Ihren Canary mit anderen AWS Diensten
- Verwenden Sie Ihren Canary dazu, eine statische IP-Adresse zu verwenden

Einen CloudWatch Synthetics-Kanarienvogel von Grund auf neu erstellen

Hier ist ein Beispiel für die Minimalversion eines Canary-Skripts für Synthetics. Dieses Skript wird als erfolgreiche Ausführung übergeben und gibt eine Zeichenfolge zurück. Um zu sehen, wie ein fehlerhaftes Canary aussieht, ändern Sie `let fail = false;` zu `let fail = true;`.

Sie müssen eine Eintrittspunkt-Funktion für das Canary-Skript definieren. Um zu sehen, wie Dateien in den Amazon-S3-Speicherort hochgeladen werden, der als `ArtifactS3Location` des Canary angegeben ist, erstellen Sie diese Dateien unter dem Ordner /tmp. Nach der Ausführung des Skripts werden der Pass/Fail-Status und die Dauer in S3 veröffentlicht CloudWatch und die Dateien unter / tmp werden auf S3 hochgeladen.

```
const basicCustomEntryPoint = async function () {

    // Insert your code here

    // Perform multi-step pass/fail check

    // Log decisions made and results to /tmp

    // Be sure to wait for all your code paths to complete
    // before returning control back to Synthetics.
    // In that way, your canary will not finish and report success
    // before your code has finished executing

    // Throw to fail, return to succeed
    let fail = false;
    if (fail) {
        throw "Failed basicCanary check.";
    }

    return "Successfully completed basicCanary checks.";
```

```
};

exports.handler = async () => {
    return await basicCustomEntryPoint();
};
```

Als Nächstes erweitern wir das Skript, um die Synthetics-Protokollierung zu verwenden und mithilfe des AWS SDK einen Anruf zu tätigen. Zu Demonstrationszwecken erstellt dieses Skript einen Amazon-DynamoDB-Client und ruft die DynamoDB-listTables-API auf. Es protokolliert die Antwort auf die Anforderung und protokolliert entweder „pass“ (erfolgreich) oder „fail“ (nicht erfolgreich), je nachdem, ob die Anforderung erfolgreich war.

```
const log = require('SyntheticsLogger');
const AWS = require('aws-sdk');
// Require any dependencies that your script needs
// Bundle additional files and dependencies into a .zip file with folder structure
// nodejs/node_modules/additional files and folders

const basicCustomEntryPoint = async function () {

    log.info("Starting DynamoDB:listTables canary.");

    let dynamodb = new AWS.DynamoDB();
    var params = {};
    let request = await dynamodb.listTables(params);
    try {
        let response = await request.promise();
        log.info("listTables response: " + JSON.stringify(response));
    } catch (err) {
        log.error("listTables error: " + JSON.stringify(err), err.stack);
        throw err;
    }

    return "Successfully completed DynamoDB:listTables canary.";
};

exports.handler = async () => {
    return await basicCustomEntryPoint();
};
```

Verpacken Sie Ihre kanarischen Dateien von Node.js

Wenn Sie Ihre Canary-Skripte über einen Amazon-S3-Speicherort hochladen, sollte Ihre ZIP-Datei Ihr Skript unter diese Ordnerstruktur enthalten: `nodejs/node_modules/`*`myCanaryFilename.js file`*.

Wenn Sie mehr als eine einzelne `.js`-Datei oder eine Abhängigkeit vorliegt, von der Ihr Skript abhängt, können Sie sie alle in einer einzigen ZIP-Datei bündeln, die die Ordnerstruktur `nodejs/node_modules/`*`myCanaryFilename.js file and other folders and files`* enthält. Wenn Sie `syn-nodejs-puppeteer-3.4` oder neuer verwenden, können Sie Ihre Canary-Dateien auch in einem anderen Ordner ablegen und Ihre Ordnerstruktur wie folgt erstellen: `nodejs/node_modules/`*`myFolder`*`/`*`myCanaryFilename.js file and other folders and files`*.

Handlername

Legen Sie den Skript-Eintrittspunkt Ihres Canary (Handler) als `myCanaryFilename.functionName` fest, damit er mit dem Dateinamen des Skript-Eintrittspunkts übereinstimmt. Wenn Sie eine Laufzeit vor `syn-nodejs-puppeteer-3.4` verwenden, muss `functionName` `handler` sein. Wenn Sie `syn-nodejs-puppeteer-3.4` oder neuer verwenden, können Sie einen beliebigen Funktionsnamen als Handler auswählen. Wenn Sie `syn-nodejs-puppeteer-3.4` oder neuer verwenden, können Sie den Canary auch in einem separaten Ordner (z. B. `nodejs/node_modules/myFolder/my_canary_filename`) ablegen. Wenn Sie ihn in einem separaten Ordner speichern, geben Sie den entsprechenden Pfad in Ihrem Skript-Eintrittspunkt an (z. B. `myFolder/my_canary_filename.functionName`).

Ändern eines vorhandenen Puppeteer-Skripts zur Verwendung als Synthetics-Canary

In diesem Abschnitt wird erläutert, wie Puppeteer-Skripte so geändert werden, dass sie als Synthetics-Canary-Skripte ausgeführt werden können. Weitere Informationen zu Puppeteer finden Sie unter Puppeteer API v1.14.0.

Wir beginnen mit diesem Puppeteer-Beispielskript:

```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://example.com');
  await page.screenshot({path: 'example.png'});
```

```
  await browser.close();
})();
```

Für die Konvertierung sind die folgenden Schritte durchzuführen:

- Erstellen und exportieren Sie eine `handler`-Funktion. Der Handler ist die Eintrittsfunktion für das Skript. Wenn Sie eine Laufzeit vor `syn-nodejs-puppeteer-3.4` verwenden, muss die Handler-Funktion den Namen `handler` haben. Wenn Sie `syn-nodejs-puppeteer-3.4` oder neuer verwenden, kann die Funktion einen beliebigen Namen haben. Dieser Name muss jedoch mit dem Namen im Skript übereinstimmen. Wenn Sie `syn-nodejs-puppeteer-3.4` oder neuer verwenden, können Sie Ihre Skripte außerdem in einem beliebigen Ordner ablegen und diesen Ordner als Teil des Handlernamens angeben.

  ```
  const basicPuppeteerExample = async function () {};

  exports.handler = async () => {
      return await basicPuppeteerExample();
  };
  ```

- Verwenden Sie die `Synthetics`-Abhängigkeit.

  ```
  var synthetics = require('Synthetics');
  ```

- Verwenden Sie die `Synthetics.getPage`-Funktion, um ein Puppeteer-`Page`-Objekt zu erhalten.

  ```
  const page = await synthetics.getPage();
  ```

  Das von der Funktion Synthetics.getPage zurückgegebene Seitenobjekt hat die Ereignisse page.on `request`, `response` und `requestfailed` für die Protokollierung instrumentiert. Synthetics richtet auch die Generierung von HAR-Dateien für Anforderungen und Antworten auf der Seite ein und fügt den Benutzeragenten-Headern der ausgehenden Anforderungen auf der Seite den Canary-ARN hinzu.

Das Skript kann nun als Synthetics-Canary ausgeführt werden. Nachfolgend finden Sie das aktualisierte Skript:

```
var synthetics = require('Synthetics');  // Synthetics dependency
```

```
const basicPuppeteerExample = async function () {
    const page = await synthetics.getPage(); // Get instrumented page from Synthetics
    await page.goto('https://example.com');
    await page.screenshot({path: '/tmp/example.png'}); // Write screenshot to /tmp
 folder
};

exports.handler = async () => {  // Exported handler function
    return await basicPuppeteerExample();
};
```

Umgebungsvariablen

Sie können Umgebungsvariablen beim Erstellen von Canaries verwenden. Auf diese Weise können Sie ein einzelnes Canaryskript schreiben und dann dieses Skript mit unterschiedlichen Werten verwenden, um schnell mehrere Canarys zu erstellen, die eine ähnliche Aufgabe haben.

Angenommen, Ihre Organisation verfügt über Endpunkte wie `prod`, `dev` und `pre-release` für die verschiedenen Phasen Ihrer Softwareentwicklung und Sie müssen Canaries erstellen, um jeden dieser Endpunkte zu testen. Sie können ein einzelnes Canary-Skript schreiben, das Ihre Software testet, und dann andere Werte für die Endpunkt-Umgebungsvariable angeben, wenn Sie die drei Canarys erstellen. Wenn Sie dann einen Canary erstellen, geben Sie das Skript und die Werte an, die für die Umgebungsvariablen verwendet werden sollen.

Die Namen von Umgebungsvariablen können Buchstaben, Zahlen und den Unterstrich enthalten. Sie müssen mit einem Buchstaben beginnen und mindestens zwei Zeichen enthalten. Die Gesamtgröße Ihrer Umgebungsvariablen darf einen Wert von 4 KB nicht überschreiten. Sie können keine reservierten Lambda-Umgebungsvariablen als Namen Ihrer Umgebungsvariablen angeben. Weitere Informationen zu reservierten Umgebungsvariablen finden Sie unter Laufzeitumgebungsvariablen.

> **Important**
>
> Die Schlüssel und Werte der Umgebungsvariablen sind nicht verschlüsselt. Speichern Sie keine sensiblen Daten darin.

Das folgende Beispielskript verwendet zwei Umgebungsvariablen. Dieses Skript ist für einen Canary bestimmt, der prüft, ob eine Webseite verfügbar ist. Es verwendet Umgebungsvariablen, um sowohl die URL, die es überprüft, als auch die verwendete CloudWatch Synthetics-Protokollebene zu parametrisieren.

Die folgende Funktion setzt `LogLevel` auf den Wert der Umgebungsvariablen `LOG_LEVEL`.

```
 synthetics.setLogLevel(process.env.LOG_LEVEL);
```

Diese Funktion setzt `URL` auf den Wert der Umgebungsvariablen `URL`.

```
const URL = process.env.URL;
```

Dies ist das komplette Skript. Wenn Sie mit diesem Skript einen Canary erstellen, geben Sie Werte für die Umgebungsvariablen `LOG_LEVEL` und `URL` an.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const pageLoadEnvironmentVariable = async function () {

    // Setting the log level (0-3)
    synthetics.setLogLevel(process.env.LOG_LEVEL);
    // INSERT URL here
    const URL = process.env.URL;

    let page = await synthetics.getPage();
    //You can customize the wait condition here. For instance,
    //using 'networkidle2' may be less restrictive.
    const response = await page.goto(URL, {waitUntil: 'domcontentloaded', timeout:
 30000});
    if (!response) {
        throw "Failed to load page!";
    }
    //Wait for page to render.
    //Increase or decrease wait time based on endpoint being monitored.
    await page.waitFor(15000);
    await synthetics.takeScreenshot('loaded', 'loaded');
    let pageTitle = await page.title();
    log.info('Page title: ' + pageTitle);
    log.debug('Environment variable:' + process.env.URL);

    //If the response status code is not a 2xx success code
    if (response.status() < 200 || response.status() > 299) {
        throw "Failed to load page!";
    }
};
```

```
exports.handler = async () => {
    return await pageLoadEnvironmentVariable();
};
```

Übergeben von Umgebungsvariablen an Ihr Skript

Um Umgebungsvariablen an Ihr Skript zu übergeben, wenn Sie einen Canary in der Konsole erstellen, geben Sie die Schlüssel und Werte der Umgebungsvariablen im Abschnitt Umgebungsvariablen der Konsole an. Weitere Informationen finden Sie unter Erstellen eines Canarys.

Um Umgebungsvariablen über die API oder zu übergeben AWS CLI, verwenden Sie den `EnvironmentVariables` Parameter im Abschnitt. `RunConfig` Im Folgenden finden Sie einen AWS CLI Beispielbefehl, der einen Canary erstellt, der zwei Umgebungsvariablen mit den Schlüsseln `Environment` und verwendet`Region`.

```
aws synthetics create-canary --cli-input-json '{
   "Name":"nameofCanary",
   "ExecutionRoleArn":"roleArn",
   "ArtifactS3Location":"s3://cw-syn-results-123456789012-us-west-2",
   "Schedule":{
      "Expression":"rate(0 minute)",
      "DurationInSeconds":604800
   },
   "Code":{
      "S3Bucket": "canarycreation",
      "S3Key": "cwsyn-mycanaryheartbeat-12345678-d1bd-1234-
abcd-123456789012-12345678-6a1f-47c3-b291-123456789012.zip",
      "Handler":"pageLoadBlueprint.handler"
   },
   "RunConfig": {
      "TimeoutInSeconds":60,
      "EnvironmentVariables": {
         "Environment":"Production",
         "Region": "us-west-1"
      }
   },
   "SuccessRetentionPeriodInDays":13,
   "FailureRetentionPeriodInDays":13,
   "RuntimeVersion":"syn-nodejs-2.0"
}'
```

Integrieren Sie Ihren Canary mit anderen AWS Diensten

Alle Kanarienvögel können die AWS SDK-Bibliothek verwenden. Du kannst diese Bibliothek verwenden, wenn du deinen Canary schreibst, um den Canary in andere AWS Dienste zu integrieren.

Dazu müssen Sie den folgenden Code zu Ihrem Canary hinzufügen. In diesen Beispielen AWS Secrets Manager wird sie als Dienst verwendet, in den der Canary integriert wird.

- Importieren Sie das AWS SDK.

```
const AWS = require('aws-sdk');
```

- Erstellen Sie einen Client für den AWS Service, in den Sie integrieren.

```
const secretsManager = new AWS.SecretsManager();
```

- Verwenden Sie den Client, um API-Aufrufe für diesen Service durchzuführen.

```
var params = {
  SecretId: secretName
};
return await secretsManager.getSecretValue(params).promise();
```

Das folgende Code-Snippet aus einem Canary-Skript veranschaulicht anhand eines Beispiels die Integration in Secrets Manager detaillierter.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const AWS = require('aws-sdk');
const secretsManager = new AWS.SecretsManager();

const getSecrets = async (secretName) => {
    var params = {
        SecretId: secretName
    };
    return await secretsManager.getSecretValue(params).promise();
}

const secretsExample = async function () {
    let URL = "<URL>";
```

```
    let page = await synthetics.getPage();

    log.info(`Navigating to URL: ${URL}`);
    const response = await page.goto(URL, {waitUntil: 'domcontentloaded', timeout:
 30000});

    // Fetch secrets
    let secrets = await getSecrets("secretname")

    /**
    * Use secrets to login.
    *
    * Assuming secrets are stored in a JSON format like:
    * {
    *   "username": "<USERNAME>",
    *   "password": "<PASSWORD>"
    * }
    **/
    let secretsObj = JSON.parse(secrets.SecretString);
    await synthetics.executeStep('login', async function () {
        await page.type(">USERNAME-INPUT-SELECTOR<", secretsObj.username);
        await page.type(">PASSWORD-INPUT-SELECTOR<", secretsObj.password);

        await Promise.all([
          page.waitForNavigation({ timeout: 30000 }),
          await page.click(">SUBMIT-BUTTON-SELECTOR<")
        ]);
    });

    // Verify login was successful
    await synthetics.executeStep('verify', async function () {
        await page.waitForXPath(">SELECTOR<", { timeout: 30000 });
    });
};

exports.handler = async () => {
    return await secretsExample();
};
```

Verwenden Sie Ihren Canary dazu, eine statische IP-Adresse zu verwenden

Sie können einen Canary so einrichten, dass er eine statische IP-Adresse verwendet.

So erzwingen Sie einen Canary dazu, eine statische IP-Adresse zu verwenden

1. Erstellen einer neuen VPC. Weitere Informationen finden Sie unter Using DNS with Your VPC.
2. Erstellen eines neuen Internet-Gateways Weitere Informationen finden Sie unter Hinzufügen eines Internet-Gateways zu Ihrer VPC.
3. Erstellen Sie ein öffentliches Subnetz in Ihrer neuen VPC.
4. Fügen Sie der VPC eine neue Routing-Tabelle hinzu.
5. Fügen Sie in der neuen Routing-Tabelle eine Route hinzu, die von `0.0.0.0/0` zum Internet-Gateway führt.
6. Verknüpfen Sie die neue Routing-Tabelle dem öffentlichen Subnetz.
7. Erstellen einer elastischen IP-Adresse Weitere Informationen finden Sie unter elastische IP-Adressen.
8. Erstellen Sie ein neues NAT-Gateway und weisen Sie es dem öffentlichen Subnetz und der elastischen IP-Adresse zu.
9. Erstellen Sie ein privates Subnetz innerhalb der VPC.
10. Fügen Sie der VPC-Standard-Routing-Tabelle eine Route hinzu, die von `0.0.0.0/0` zum NAT-Gateway geht
11. Erstellen Sie Ihr Canary.

Ein Python-Canary-Skript schreiben

Dieses Skript wird als erfolgreiche Ausführung übergeben und gibt eine Zeichenfolge zurück. Um zu sehen, wie ein fehlgeschlagener Canary aussieht, ändern Sie Fail = False in Fail = True

```
def basic_custom_script():
    # Insert your code here
    # Perform multi-step pass/fail check
    # Log decisions made and results to /tmp
    # Be sure to wait for all your code paths to complete
    # before returning control back to Synthetics.
    # In that way, your canary will not finish and report success
    # before your code has finished executing
    fail = False
    if fail:
        raise Exception("Failed basicCanary check.")
    return "Successfully completed basicCanary checks."
def handler(event, context):
```

```
    return basic_custom_script()
```

Verpacken Sie Ihre kanarischen Python-Dateien

Wenn Sie mehr als eine .py-Datei haben oder Ihr Skript eine Abhängigkeit aufweist, können Sie sie alle in einer einzigen ZIP-Datei bündeln. Wenn Sie die `syn-python-selenium-1.1`-Laufzeit verwenden, muss die ZIP-Datei Ihre canary.py-Datei in einem `python`-Ordner enthalten (z. B. `python/my_canary_filename.py`). Wenn Sie `syn-python-selenium-1.1` oder neuer verwenden, können Sie auch einen anderen Ordner nutzen, etwa `python/myFolder/my_canary_filename.py`.

Diese ZIP-Datei sollte alle erforderlichen Ordner und Dateien enthalten, die anderen Dateien müssen sich jedoch nicht im `python`-Ordner befinden.

Legen Sie den Skript-Eintrittspunkt Ihres Canary als `my_canary_filename.functionName` fest, damit er mit dem Datei- und Funktionsnamen des Skript-Eintrittspunkts übereinstimmt. Wenn Sie die Laufzeit `syn-python-selenium-1.0` verwenden, muss `functionName` `handler` sein. Wenn Sie `syn-python-selenium-1.1` oder neuer verwenden, gilt diese Einschränkung für den Handlernamen nicht und Sie können den Canary auch in einem separaten Ordner (z. B. `python/myFolder/my_canary_filename.py`) ablegen. Wenn Sie ihn in einem separaten Ordner speichern, geben Sie den entsprechenden Pfad in Ihrem Skript-Eintrittspunkt an (z. B. `myFolder/my_canary_filename.functionName`).

Ändern eines vorhandenen Selenium-Skripts zur Verwendung eines Synthetics-Canarys

Sie können schnell ein vorhandenes Skript für Python und Selenium ändern, um als Canary verwendet zu werden. Weitere Informationen zu Selenium finden Sie unter www.selenium.dev/.

In diesem Beispiel beginnen wir mit dem folgenden Selenium-Skript:

```
from selenium import webdriver

def basic_selenium_script():
    browser = webdriver.Chrome()
    browser.get('https://example.com')
    browser.save_screenshot('loaded.png')

basic_selenium_script()
```

Für die Konvertierung sind die folgenden Schritte durchzuführen.

So konvertieren Sie ein Selenium-Skript für die Verwendung als Canary

1. Ändern Sie die `import`-Anweisung, um Selenium aus dem `aws_synthetics`-Modul zu verwenden:

   ```
   from aws_synthetics.selenium import synthetics_webdriver as webdriver
   ```

   Das Selenium-Modul von `aws_synthetics` stellt sicher, dass der Canary Metriken und Protokolle ausgeben, eine HAR-Datei generieren und mit anderen CloudWatch Synthetics-Funktionen arbeiten kann.

2. Erstellen Sie eine Handler-Funktion und rufen Sie Ihre Selenium-Methode auf. Der Handler ist die Eintrittsfunktion für das Skript.

   Wenn Sie `syn-python-selenium-1.0` verwenden, muss die Handler-Funktion den Namen `handler` haben. Wenn Sie `syn-python-selenium-1.1` oder neuer verwenden, kann die Funktion einen beliebigen Namen haben. Dieser Name muss jedoch mit dem Namen im Skript übereinstimmen. Wenn Sie `syn-python-selenium-1.1` oder neuer verwenden, können Sie Ihre Skripte außerdem in einem beliebigen Ordner ablegen und diesen Ordner als Teil des Handlernamens angeben.

   ```
   def handler(event, context):
       basic_selenium_script()
   ```

Das Skript wurde jetzt aktualisiert und ist jetzt ein CloudWatch Synthetics-Kanarienvogel. Nachfolgend finden Sie das aktualisierte Skript:

```
from aws_synthetics.selenium import synthetics_webdriver as webdriver

def basic_selenium_script():
    browser = webdriver.Chrome()
    browser.get('https://example.com')
    browser.save_screenshot('loaded.png')

def handler(event, context):
    basic_selenium_script()
```

Änderung eines vorhandenen Puppeteer Synthetics-Skripts zur Authentifizierung von nicht standardmäßigen Zertifikaten

Ein wichtiger Anwendungsfall für Synthetics Canaries ist die Überwachung Ihrer eigenen Endgeräte. Wenn Sie einen Endpunkt überwachen möchten, der nicht für externen Datenverkehr bereit ist, kann diese Überwachung manchmal bedeuten, dass Sie kein richtiges Zertifikat haben, das von einer vertrauenswürdigen Zertifizierungsstelle eines Drittanbieters signiert wurde.

Zwei mögliche Lösungen für dieses Szenario lauten wie folgt:

- Informationen zur Authentifizierung eines Client-Zertifikats finden Sie unter So validieren Sie die Authentifizierung mit Amazon CloudWatch Synthetics — Teil 2.
- Informationen zur Authentifizierung eines selbstsignierten Zertifikats finden Sie unter So validieren Sie die Authentifizierung mit selbstsignierten Zertifikaten in Amazon Synthetics CloudWatch

Sie sind nicht auf diese beiden Optionen beschränkt, wenn Sie CloudWatch Synthetics Canaries verwenden. Sie können diese Funktionen erweitern und Ihre Geschäftslogik hinzufügen, indem Sie den Canary-Code erweitern.

> **Note**
>
> Synthetics Canaries, die auf Python-Laufzeiten laufen, haben das `--ignore-certificate-errors` Flag von Haus aus aktiviert, sodass diese Canaries keine Probleme haben sollten, Websites mit nicht standardmäßigen Zertifikatkonfigurationen zu erreichen.

Für Canary-Skripte verfügbare Bibliotheksfunktionen

CloudWatch Synthetics enthält mehrere integrierte Klassen und Funktionen, die Sie aufrufen können, wenn Sie Node.js -Skripten schreiben, die als Kanariendateien verwendet werden sollen.

Einige gelten sowohl für Benutzeroberflächen-(UI)- als auch für API-Canarys. Andere gelten nur für UI-Canarys. Ein UI-Canary ist ein Canary, das die Funktion `getPage()` verwendet und Puppeteer als Webtreiber verwendet, um zu Webseiten zu navigieren und mit ihnen zu interagieren.

> **Note**
>
> Wenn Sie ein Canary auf eine neue Version der Synthetics-Laufzeitumgebung aktualisieren, werden auch alle Synthetics-Bibliotheksfunktionen, die Ihr Canary verwendet, automatisch

auf die gleiche Version von NodeJS aktualisiert, die die Synthetics-Laufzeitumgebung unterstützt.

Themen

- Für Node.js-Canary-Skripte verfügbare Bibliotheksfunktionen
- Verfügbare Bibliotheksfunktionen für Python-Canary-Skripte mit Selenium

Für Node.js-Canary-Skripte verfügbare Bibliotheksfunktionen

In diesem Abschnitt werden die Bibliotheksfunktionen aufgelistet, die für Canary-Skripts von Node.js verfügbar sind.

Themen

- Node.js-Bibliotheksfunktionen, die für alle Canarys gelten
- Node.js-Bibliotheksklassen und -funktionen, die nur für UI-Canaries gelten
- Node.js-Bibliotheksklassen und -funktionen, die nur für API-Canaries gelten

Node.js-Bibliotheksfunktionen, die für alle Canarys gelten

Die folgenden CloudWatch Synthetics-Bibliotheksfunktionen für Node.js sind für alle Kanarienvögel nützlich.

Themen

- Synthetics-Klasse
- SyntheticsConfiguration Klasse
- Synthetics Logger
- SyntheticsLogHelper Klasse

Synthetics-Klasse

Die folgenden Funktionen für alle Canarys befinden sich in der Klasse Synthetics.

addExecutionError(errorMessage, ex);

`errorMessage` beschreibt den Fehler und `ex` ist die aufgetretene Ausnahme

Sie können Folgendes verwenden:`addExecutionError`, um Ausführungsfehler für Ihren Canary festzulegen. Es lässt den Canary fehlschlagen, ohne die Skriptausführung zu unterbrechen. Es wirkt sich auch nicht auf Ihre `successPercent`-Metriken aus.

Sie sollten Fehler nur dann als Ausführungsfehler verfolgen, wenn sie nicht wichtig sind, um den Erfolg oder Misserfolg Ihres Canary-Skripts anzuzeigen.

Ein Beispiel für die Verwendung von `addExecutionError` ist das folgende. Sie überwachen die Verfügbarkeit Ihres Endpunkts und machen Screenshots, nachdem die Seite geladen wurde. Da der Fehler beim Erstellen eines Screenshots die Verfügbarkeit des Endpunkts nicht bestimmt, können Sie beim Erstellen von Screenshots aufgetretene Fehler abfangen und sie als Ausführungsfehler hinzufügen. Ihre Verfügbarkeitsmetriken zeigen weiterhin an, dass der Endpunkt aktiv ist und ausgeführt wird, aber Ihr Canary-Status wird als fehlgeschlagen markiert. Der folgende Codeblock fängt einen solchen Fehler ab und fügt ihn als Ausführungsfehler hinzu.

```
try {
        await synthetics.takeScreenshot(stepName, "loaded");
    } catch(ex) {
        synthetics.addExecutionError('Unable to take screenshot ', ex);
    }
```

getCanaryName();

Gibt den Namen des Canarys zurück.

getCanaryArn();

Gibt den ARN des Canarys aus.

getCanaryUserAgentString();

Gibt den benutzerdefinierten Benutzeragenten des Canary zurück.

getRuntimeVersion();

Diese Funktion ist in Laufzeitversion `syn-nodejs-puppeteer-3.0` und höher verfügbar. Es gibt die Synthetics Laufzeitversion des Canarys zurück. Der Rückgabewert könnte beispielsweise `syn-nodejs-puppeteer-3.0` sein.

getLogLevel();

Ruft die aktuelle Protokollebene für die Synthetics-Bibliothek ab. Folgende Werte sind möglich:

- `0` – Debug
- `1` – Info
- `2` – Warnen
- `3` – Fehler

Beispiel:

```
let logLevel = synthetics.getLogLevel();
```

setLogLevel();

Legt die Protokollebene für die Synthetics-Bibliothek fest. Folgende Werte sind möglich:

- `0` – Debug
- `1` – Info
- `2` – Warnen
- `3` – Fehler

Beispiel:

```
synthetics.setLogLevel(0);
```

SyntheticsConfiguration Klasse

Diese Klasse ist nur in der `syn-nodejs-2.1`-Laufzeitversion oder höher verfügbar.

Sie können die SyntheticsConfiguration Klasse verwenden, um das Verhalten von Synthetics-Bibliotheksfunktionen zu konfigurieren. Sie können diese Klasse beispielsweise verwenden, um die `executeStep()`-Funktion so zu konfigurieren, dass keine Screenshots erfasst werden.

Sie können CloudWatch Synthetics-Konfigurationen auf globaler Ebene festlegen, die auf alle Stufen der Kanaren angewendet werden. Sie können diese Konfigurationen auch auf Schrittebene überschreiben, indem Sie Konfigurationsschlüssel-Wert-Paare übergeben.

Sie können Optionen auf Schrittebene übergeben. Beispiele finden Sie unter asynchron executeStep (stepName, functionToExecute, [StepConfig]); und executeHttpStep(stepName, RequestOptions, [Rückruf], [StepConfig]).

Funktionsdefinitionen:

setConfig (Optionen)

`options` ist ein Objekt, bei dem es sich um eine Reihe konfigurierbarer Optionen für Ihren Canary handelt. In den folgenden Abschnitten werden die möglichen Felder in *`options`* erläutert.

setConfig (Optionen) für alle Canary

Für Canaries, die `syn-nodejs-puppeteer-3.2` oder höher verwenden, können die (Optionen) für setConfig die folgenden Parameter enthalten:

- `includeRequestHeaders` (boolean) – Gibt an, ob Anforderungs-Header in den Bericht aufgenommen werden sollen. Der Standardwert ist `false`.
- `includeResponseHeaders` (boolean) – Gibt an, ob Antwort-Header in den Bericht aufgenommen werden sollen. Der Standardwert ist `false`.
- `restrictedHeaders` (array) – Eine Liste von Header-Werten, die ignoriert werden sollen, wenn Header enthalten sind. Dies gilt sowohl für Anforderungs- als auch für Antwort-Header. Sie können Ihre Anmeldeinformationen beispielsweise verbergen, indem Sie includeRequestHeadersas `true` und RestrictedHeaders als übergeben. `['Authorization']`
- `includeRequestBody` (boolean) – Gibt an, ob der Anforderungstext in den Bericht aufgenommen werden soll. Der Standardwert ist `false`.
- `includeResponseBody` (boolean) – Gibt an, ob der Antworttext in den Bericht aufgenommen werden soll. Der Standardwert ist `false`.

setConfig (Optionen) in Bezug auf Metriken CloudWatch

Für Canarys, die `syn-nodejs-puppeteer-3.1` oder höher verwenden, können die (Optionen) für setConfig die folgenden booleschen Parameter enthalten, die bestimmen, welche Metriken vom Canary veröffentlicht werden. Der Standardwert für jede dieser Optionen ist `true`. Die Optionen, die mit `aggregated` beginnen, bestimmen, ob die Metrik ohne die `CanaryName`-Dimension ausgegeben wird. Sie können diese Metriken verwenden, um die aggregierten Ergebnisse für alle Canarys anzuzeigen. Die anderen Optionen bestimmen, ob die Metrik mit der `CanaryName` Dimension ausgegeben wird. Sie können diese Metriken verwenden, um die Ergebnisse für jeden einzelnen Canary anzuzeigen.

Eine Liste der von Canaries ausgegebenen CloudWatch Messwerte finden Sie unter. CloudWatch von Canaries veröffentlichte Metriken

- `failedCanaryMetric` (boolean) – Gibt an, ob die `Failed`-Metrik (mit der `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `failedRequestsMetric` (boolean) – Gibt an, ob die `Failed requests`-Metrik (mit der `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `_2xxMetric` (boolean) – Gibt an, ob die `2xx`-Metrik (mit der `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `_4xxMetric` (boolean) – Gibt an, ob die `4xx`-Metrik (mit der `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `_5xxMetric` (boolean) – Gibt an, ob die `5xx`-Metrik (mit der `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `stepDurationMetric` (boolean) – Gibt an, ob die `Step duration`-Metrik (mit der `CanaryName`-Dimension `StepName`) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `stepSuccessMetric` (boolean) – Gibt an, ob die `Step success`-Metrik (mit der `CanaryName`-Dimension `StepName`) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `aggregatedFailedCanaryMetric` (boolean) – Gibt an, ob die `Failed`-Metrik (ohne die `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `aggregatedFailedRequestsMetric` (boolean) – Gibt an, ob die `Failed Requests`-Metrik (ohne die `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `aggregated2xxMetric` (boolean) – Gibt an, ob die `2xx`-Metrik (ohne die `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `aggregated4xxMetric` (boolean) – Gibt an, ob die `4xx`-Metrik (ohne die `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `aggregated5xxMetric` (boolean) – Gibt an, ob die `5xx`-Metrik (ohne die `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `visualMonitoringSuccessPercentMetric` (boolean) – Gibt an, ob die `visualMonitoringSuccessPercent`-Metrik für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `visualMonitoringTotalComparisonsMetric` (boolean) – Gibt an, ob die `visualMonitoringTotalComparisons`-Metrik für diesen Canary emittiert werden soll. Der Standardwert ist `false`.

- `stepsReport` (boolean) – Gibt an, ob eine Zusammenfassung der Schrittausführung gemeldet werden soll. Der Standardwert ist `true`.
- `includeUrlPassword` (boolean) – Gibt an, ob ein Passwort eingefügt werden soll, das in der URL angezeigt wird. Standardmäßig werden Passwörter, die in URLs angezeigt werden, aus Protokollen und Berichten abgedichtet, um die Offenlegung vertraulicher Daten zu verhindern. Der Standardwert ist `false`.
- `restrictedUrlParameters` (array) – Eine Liste von URL-Pfad- oder Abfrageparametern, die geschwärzt werden sollen. Dies gilt für URLs, die in Protokollen, Berichten und Fehlern angezeigt werden. Bei dem Parameter wird die Groß-/Kleinschreibung nicht beachtet. Sie können ein Sternchen (*) als Wert übergeben, um alle URL-Pfad- und Abfrageparameterwerte zu überarbeiten. Der Standardwert ist ein leeres Array.
- `logRequest` (boolean) – Gibt an, ob jede Anforderung in Canary-Protokollen protokolliert werden soll. Bei UI-Canarys protokolliert dies jede Anforderung, die vom Browser gesendet wird. Der Standardwert ist `true`.
- `logResponse` (boolean) – Gibt an, ob jede Antwort in Canary-Protokollen protokolliert werden soll. Bei UI-Canarys protokolliert dies jede vom Browser empfangene Antwort. Der Standardwert ist `true`.
- `logRequestBody` (boolean) – Gibt an, ob Anforderungstexte zusammen mit den Anforderungen in Canary- Protokollen protokolliert werden sollen. Diese Konfiguration gilt nur, wenn `logRequest` `true` ist. Der Standardwert ist `false`.
- `logResponseBody` (boolean) – Gibt an, ob Antworttexte zusammen mit den Antworten in Canary-Logs protokolliert werden sollen. Diese Konfiguration gilt nur, wenn `logResponse` `true` ist. Der Standardwert ist `false`.
- `logRequestHeaders` (boolean) – Gibt an, ob Anforderungsheader zusammen mit den Anforderungen in Canary-Protokollen protokolliert werden sollen. Diese Konfiguration gilt nur, wenn `logRequest` `true` ist. Der Standardwert ist `false`.

  Beachten Sie, dass `includeRequestHeaders` Header in Artefakten aktiviert.
- `logResponseHeaders` (boolean) – Gibt an, ob Antwortheader zusammen mit den Antworten in Canary-Protokollen protokolliert werden sollen. Diese Konfiguration gilt nur, wenn `logResponse` `true` ist. Der Standardwert ist `false`.

  Beachten Sie, dass `includeResponseHeaders` Header in Artefakten aktiviert.

> **Note**
>
> Die `Duration`- und `SuccessPercent`-Metriken werden immer für jeden Canary ausgegeben, sowohl mit als auch ohne die `CanaryName`-Metrik

Methoden zum Aktivieren oder Deaktivieren von Metriken

disableAggregatedRequestMetriken ()

Verhindert, dass der Canary alle Anforderungsmesswerte ausgibt, die ohne `CanaryName`-Dimension ausgegeben werden.

disableRequestMetrics()

Deaktiviert alle Anforderungsmetriken, einschließlich aller Canary-Metriken und Metriken, die über alle Canarys aggregiert werden.

disableStepMetrics()

Deaktiviert alle Schrittmetriken, einschließlich Metriken für den Schritterfolg und für die Schrittdauer.

enableAggregatedRequestMetriken ()

Ermöglicht dem Canary, alle Anforderungsmesswerte auszugeben, die ohne `CanaryName`-Dimension ausgegeben werden.

enableRequestMetrics()

Aktiviert alle Anforderungsmetriken, einschließlich aller Canary-Metriken und Metriken, die über alle Canarys aggregiert werden.

enableStepMetrics()

Aktiviert alle Schrittmetriken, einschließlich Metriken für den Schritterfolg und für die Schrittdauer.

get2xxMetric()

Gibt zurück, ob der Canary eine `2xx`-Metrik mit der `CanaryName`-Dimension ausgibt.

get4xxMetric()

Gibt zurück, ob der Canary eine `4xx`-Metrik mit der `CanaryName`-Dimension ausgibt.

get5xxMetric()

Gibt zurück, ob der Canary eine `5xx`-Metrik mit der `CanaryName`-Dimension ausgibt.

getAggregated2xxMetric()

Gibt zurück, ob der Canary eine `2xx`-Metrik ohne Dimension ausgibt.

getAggregated4xxMetric()

Gibt zurück, ob der Canary eine `4xx`-Metrik ohne Dimension ausgibt.

getAggregatedFailedCanaryMetric()

Gibt zurück, ob der Canary eine `Failed`-Metrik ohne Dimension ausgibt.

getAggregatedFailedRequestsMetric()

Gibt zurück, ob der Canary eine `Failed requests`-Metrik ohne Dimension ausgibt.

getAggregated5xxMetric()

Gibt zurück, ob der Canary eine `5xx`-Metrik ohne Dimension ausgibt.

getFailedCanaryMetrisch ()

Gibt zurück, ob der Canary eine `Failed`-Metrik mit der `CanaryName`-Dimension ausgibt.

getFailedRequestsMetrisch ()

Gibt zurück, ob der Canary eine `Failed requests`-Metrik mit der `CanaryName`-Dimension ausgibt.

getStepDurationMetrisch ()

Gibt zurück, ob der Canary eine `Duration`-Metrik mit der `CanaryName`-Dimension für dieses Canary ausgibt.

getStepSuccessMetrisch ()

Gibt zurück, ob der Canary eine `StepSuccess`-Metrik mit der `CanaryName`-Dimension für dieses Canary ausgibt.

with2xxMetric(_2xxMetric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `2xx`-Metrik mit der `CanaryName`-Dimension für diesen Canary emittiert werden soll.

with4xxMetric(_4xxMetric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `4xx`-Metrik mit der `CanaryName`-Dimension für diesen Canary emittiert werden soll.

with5xxMetric(_5xxMetric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `5xx`-Metrik mit der `CanaryName`-Dimension für diesen Canary emittiert werden soll.

withAggregated2xxMetric(aggregated2xxMetric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `2xx`-Metrik mit keiner Dimension für diesen Canary emittiert werden soll.

withAggregated4xxMetric(aggregated4xxMetric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `4xx`-Metrik mit keiner Dimension für diesen Canary emittiert werden soll.

withAggregated5xxMetric(aggregated5xxMetric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `5xx`-Metrik mit keiner Dimension für diesen Canary emittiert werden soll.

withAggregatedFailedCanaryMetric(aggregatedFailedCanaryMetrisch)

Akzeptiert ein boolesches Argument, das angibt, ob eine `Failed`-Metrik mit keiner Dimension für diesen Canary emittiert werden soll.

withAggregatedFailedRequestsMetric(aggregatedFailedRequestsMetrisch)

Akzeptiert ein boolesches Argument, das angibt, ob eine `Failed requests`-Metrik mit keiner Dimension für diesen Canary emittiert werden soll.

withFailedCanaryMetrisch (failedCanaryMetric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `Failed`-Metrik mit der `CanaryName`-Dimension für diesen Canary emittiert werden soll.

withFailedRequestsMetrisch (failedRequestsMetric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `Failed requests`-Metrik mit der `CanaryName`-Dimension für diesen Canary emittiert werden soll.

withStepDurationMetrisch (stepDurationMetric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `Duration`-Metrik mit der `CanaryName`-Dimension für diesen Canary emittiert werden soll.

withStepSuccessMetrisch (stepSuccessMetric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `StepSuccess`-Metrik mit der `CanaryName`-Dimension für diesen Canary emittiert werden soll.

Methoden zum Aktivieren oder Deaktivieren anderer Features

withHarFile()

Akzeptiert ein boolesches Argument, das angibt, ob eine HAR-Datei für diesen Canary erstellt werden soll.

withStepsReport()

Akzeptiert ein boolesches Argument, das angibt, ob eine Zusammenfassung der Schrittausführung für diesen Canary gemeldet werden soll.

withIncludeUrlPasswort ()

Akzeptiert ein boolesches Argument, das angibt, ob Kennwörter eingeschlossen werden sollen, die in URLs in Protokollen und Berichten angezeigt werden.

withRestrictedUrlParameter ()

Akzeptiert ein Array von URL-Pfad- oder Abfrageparametern zum Schwärzen. Dies gilt für URLs, die in Protokollen, Berichten und Fehlern angezeigt werden. Sie können ein Sternchen (*) als Wert übergeben, um alle URL-Pfad- und Abfrageparameterwerte zu verkleinern

withLogRequest()

Akzeptiert ein boolesches Argument, das angibt, ob jede Anforderung in den Protokollen des Canarys protokolliert werden soll.

withLogResponse()

Akzeptiert ein boolesches Argument, das angibt, ob jede Antwort in den Protokollen des Canarys protokolliert werden soll.

withLogRequestKörper ()

Akzeptiert ein boolesches Argument, das angibt, ob jeder Anforderungstext in den Protokollen des Canarys protokolliert werden soll.

withLogResponseKörper ()

Akzeptiert ein boolesches Argument, das angibt, ob jeder Antworttext in den Protokollen des Canarys protokolliert werden soll.

withLogRequestÜberschriften ()

Akzeptiert ein boolesches Argument, das angibt, ob jeder Anforderungs-Header in den Protokollen des Canarys protokolliert werden soll.

withLogResponseÜberschriften ()

Akzeptiert ein boolesches Argument, das angibt, ob jeder Antwort-Header in den Protokollen des Canarys protokolliert werden soll.

getHarFile()

Gibt zurück, ob der Canary eine HAR-Datei erstellt.

getStepsReport()

Gibt zurück, ob der Canary eine Zusammenfassung der Schrittausführung meldet

getIncludeUrlPasswort ()

Gibt zurück, ob der Canary Passwörter enthält, die in URLs in Protokollen und Berichten angezeigt werden.

getRestrictedUrlParameter ()

Gibt zurück, ob der Canary den URL-Pfad oder die Abfrageparameter entfernt.

getLogRequest()

Gibt zurück, ob der Canary jede Anforderung in den Canaryprotokollen protokolliert.

getLogResponse()

Gibt zurück, ob der Canary jede Antwort in den Canaryprotokollen protokolliert.

getLogRequestKörper ()

Gibt zurück, ob der Canary jeden Anforderungstext in den Canaryprotokollen protokolliert.

getLogResponseKörper ()

Gibt zurück, ob der Canary jeden Antworttext in den Canaryprotokollen protokolliert.

getLogRequestÜberschriften ()

Gibt zurück, ob der Canary jeden Anforderungsheader in den Protokollen des Canarys.

getLogResponseÜberschriften ()

Gibt zurück, ob der Canary alle Antwort-Header in den Canaryprotokollen protokolliert.

Funktionen für alle Canarys

- `withIncludeRequestHeaders`(includeRequestHeaders)
- `withIncludeResponseHeaders`(includeResponseHeaders)
- `withRestrictedHeaders`(RestrictedHeader)
- `withIncludeRequestBody`(includeRequestBody)
- `withIncludeResponseBody`(includeResponseBody)
- `enableReportingOptions`() — Aktiviert alle Berichtsoptionen-- includeRequestHeaders, includeResponseHeadersincludeRequestBody, und includeResponseBody,.
- `disableReportingOptions`() — Deaktiviert alle Berichtsoptionen-- includeRequestHeaders, includeResponseHeadersincludeRequestBody, und includeResponseBody,.

setConfig (Optionen) für UI-Canarys

Für UI-Canarys kann SetConfig folgende boolesche Parameter enthalten:

- `continueOnStepFailure` (boolean) - Gibt an, b das Canary-Skript nach einem fehlgeschlagenen Schritt weiter ausgeführt werden soll (dies bezieht sich auf die Funktion executeStep). Wenn Schritte fehlschlagen, wird der Canary-Lauf weiterhin als fehlgeschlagen markiert. Der Standardwert ist `false`.
- `harFile` (boolean) – Gibt an, ob eine HAR-Datei erstellt werden soll. Der Standardwert ist `True`.
- `screenshotOnStepStart` (boolean) – Gibt an, ob ein Screenshot erstellt werden soll, bevor ein Schritt gestartet wird.

- `screenshotOnStepSuccess` (boolean) – Gibt an, ob nach einem erfolgreichen Schritt ein Screenshot erstellt werden soll.
- `screenshotOnStepFailure` (boolean) - Ob ein Screenshot erstellt werden soll, nachdem ein Schritt fehlgeschlagen ist.

Methoden zum Aktivieren oder Deaktivieren von Screenshots

disableStepScreenshots()

Deaktiviert alle Screenshot-Optionen (screenshotOnStepStart, screenshotOnStep Erfolg und screenshotOnStep Fehler).

enableStepScreenshots()

Aktiviert alle Screenshot-Optionen (screenshotOnStepStart, screenshotOnStep Erfolg und screenshotOnStep Fehler). Standardmäßig sind alle diese Methoden aktiviert.

getScreenshotOnStepFailure()

Gibt zurück, ob der Canary einen Screenshot macht, nachdem ein Schritt fehlschlägt.

getScreenshotOnStepStart()

Gibt zurück, ob der Canary einen Screenshot erstellt, bevor er einen Schritt startet.

getScreenshotOnStepSuccess()

Gibt zurück, ob der Canary nach erfolgreichem Abschluss eines Schritts einen Screenshot erstellt.

withScreenshotOnStepStart(screenshotOnStepStart)

Akzeptiert ein boolesches Argument, das angibt, ob ein Screenshot erstellt werden soll, bevor ein Schritt gestartet wird.

withScreenshotOnStepSuccess(screenshotOnStepErfolg)

Akzeptiert ein boolesches Argument, das angibt, ob nach erfolgreichem Abschluss eines Schritts ein Screenshot erstellt werden soll.

withScreenshotOnStepFailure(screenshotOnStepMisserfolg)

Akzeptiert ein boolesches Argument, das angibt, ob nach einem Schritt ein Screenshot erstellt werden soll.

Verwendung in UI-Canarys

Importieren Sie zuerst die Synthetics-Abhängigkeit und holen Sie die Konfiguration ab.

```
// Import Synthetics dependency
const synthetics = require('Synthetics');

// Get Synthetics configuration
const synConfig = synthetics.getConfiguration();
```

Legen Sie dann die Konfiguration für jede Option fest, indem Sie die setConfig-Methode mit einer der folgenden Optionen aufrufen.

```
// Set configuration values
    synConfig.setConfig({
        screenshotOnStepStart: true,
        screenshotOnStepSuccess: false,
        screenshotOnStepFailure: false
    });
```

Oder

```
synConfig.withScreenshotOnStepStart(false).withScreenshotOnStepSuccess(true).withScreenshotOnSt
```

Um alle Screenshots zu deaktivieren, verwenden Sie die Funktion disableStepScreenshots () wie in diesem Beispiel.

```
synConfig.disableStepScreenshots();
```

Sie können Screenshots jederzeit im Code aktivieren oder deaktivieren. Wenn Sie beispielsweise Screenshots nur für einen Schritt deaktivieren möchten, deaktivieren Sie sie, bevor Sie diesen Schritt ausführen, und aktivieren Sie sie dann nach dem Schritt.

setConfig (Optionen) für API-Canarys

Für API-Canarys kann SetConfig folgende boolesche Parameter enthalten:

- `continueOnHttpStepFailure`(boolean) — Ob mit der Ausführung des Canary-Skripts fortgefahren werden soll, nachdem ein HTTP-Schritt fehlgeschlagen ist (dies bezieht sich auf die executeHttpStepFunktion). Wenn Schritte fehlschlagen, wird der Canary-Lauf weiterhin als fehlgeschlagen markiert. Der Standardwert ist `true`.

Visuelle Überwachung

Die visuelle Überwachung vergleicht Screenshots, die während eines Canary-Laufs aufgenommen wurden, mit Screenshots, die während eines Baseline-Canary-Laufs aufgenommen wurden. Wenn die Diskrepanz zwischen den beiden Screenshots einen Schwellenwert überschreitet, schlägt der Canary fehl, und Sie können die Bereiche mit Unterschieden in der Farbe im Canarylauf-Bericht sehen. Die visuelle Überwachung wird auf Kanaren unterstützt, auf denen die Version syn-puppeteer-node-3.2 und höher ausgeführt wird. Es wird derzeit nicht in Canarys unterstützt, die Python und Selenium ausführen.

Um die visuelle Überwachung zu aktivieren, fügen Sie dem Canary-Skript die folgende Codezeile hinzu. Weitere Details finden Sie unter SyntheticsConfiguration Klasse.

```
syntheticsConfiguration.withVisualCompareWithBaseRun(true);
```

Wenn der Canary zum ersten Mal erfolgreich ausgeführt wird, nachdem diese Zeile zum Skript hinzugefügt wurde, verwendet er die während dieser Ausführung erstellten Screenshots als Vergleichsbasis. Nach dem ersten Canary-Run kannst du die CloudWatch Konsole verwenden, um den Canary zu bearbeiten, um einen der folgenden Schritte auszuführen:

- Legen Sie den nächsten Lauf des Canarys als neue Basislinie fest.
- Zeichnen Sie Grenzen auf dem aktuellen Baseline-Screenshot, um Bereiche des Screenshots festzulegen, die bei visuellen Vergleichen ignoriert werden sollen.
- Entfernen Sie einen Screenshot, der für die visuelle Überwachung verwendet wird.

Weitere Informationen zur Verwendung der CloudWatch Konsole zum Bearbeiten eines Canary finden Sie unterEinen Canary bearbeiten oder löschen.

Weitere Optionen für die visuelle Überwachung

Synthetics-Konfiguration. withVisualVarianceThresholdPercentage(Gewünschter Prozentsatz)

Legen Sie den akzeptablen Prozentsatz für die Screenshot-Varianz in visuellen Vergleichen fest.

Konfiguration „Synthetik“. withVisualVarianceHighlightHexColor(“ #fafa00 „)

Legen Sie die Hervorhebungsfarbe fest, die Varianzbereiche angibt, wenn Sie Canary-Lauf-Berichte betrachten, die visuelle Überwachung verwenden.

Synthetische Konfiguration. withFailCanaryRunOnVisualVariance(Canary scheitern)

Legen Sie fest, ob der Canary fehlschlägt, wenn ein visueller Unterschied größer als der Schwellenwert ist. Die Standardeinstellung ist, dass der Canary fehlschlägt.

Synthetics Logger

SyntheticsLogger schreibt Logs sowohl in die Konsole als auch in eine lokale Protokolldatei auf derselben Protokollebene. Diese Protokolldatei wird nur dann an beide Speicherorte geschrieben, wenn die Protokollebene auf oder unter der gewünschten Protokollierungsebene der aufgerufenen Protokollfunktion liegt.

Den Protokollierungsanweisungen in der lokalen Protokolldatei werden je nach der Protokollebene der aufgerufenen Funktion „DEBUG“ „INFO“ usw. vorangestellt.

Sie können das verwenden SyntheticsLogger, vorausgesetzt, Sie möchten die Synthetics Library auf derselben Protokollebene wie Ihr Synthetics Canary-Logging ausführen.

Die Verwendung von SyntheticsLogger ist nicht erforderlich, um eine Protokolldatei zu erstellen, die an Ihren S3-Ergebnisspeicherort hochgeladen wird. Sie können stattdessen eine andere Protokolldatei im `/tmp`-Ordner erstellen. Alle Dateien, die unter dem `/tmp`-Ordner erstellt wurden, werden als Artefakte an den Ergebnisspeicherort in S3 hochgeladen.

So verwenden Sie den Synthetics Library Logger:

```
const log = require('SyntheticsLogger');
```

Nützliche Funktionsdefinitionen:

log.debug(***message***, ***ex***);

Parameter: *message* ist die zu protokollierende Nachricht. *ex* ist die eventuell zu protokollierende Ausnahme.

Beispiel:

```
log.debug("Starting step - login.");
```

log.error(***message***, ***ex***);

Parameter: *message* ist die zu protokollierende Nachricht. *ex* ist die eventuell zu protokollierende Ausnahme.

Beispiel:

```
try {
  await login();
catch (ex) {
  log.error("Error encountered in step - login.", ex);
}
```

log.info(***message***, ***ex***);

Parameter: *message* ist die zu protokollierende Nachricht. *ex* ist die eventuell zu protokollierende Ausnahme.

Beispiel:

```
log.info("Successfully completed step - login.");
```

log.log(***message***, ***ex***);

Dies ist ein Alias für `log.info`.

Parameter: *message* ist die zu protokollierende Nachricht. *ex* ist die eventuell zu protokollierende Ausnahme.

Beispiel:

```
 log.log("Successfully completed step - login.");
```

log.warn(***message***, ***ex***);

Parameter: *message* ist die zu protokollierende Nachricht. *ex* ist die eventuell zu protokollierende Ausnahme.

Beispiel:

```
log.warn("Exception encountered trying to publish CloudWatch Metric.", ex);
```

SyntheticsLogHelper Klasse

Die Klasse `SyntheticsLogHelper` ist in der Laufzeit `syn-nodejs-puppeteer-3.2` und späteren Laufzeiten verfügbar. Es ist bereits in der CloudWatch Synthetics-Bibliothek initialisiert und mit der Synthetics-Konfiguration konfiguriert. Sie können dies in Ihrem Skript als Abhängigkeit

hinzufügen. Diese Klasse ermöglicht es Ihnen, URLs, Header und Fehlermeldungen zu bereinigen, um vertrauliche Informationen zu überarbeiten.

> **Note**
>
> Synthetics bereinigt alle protokollierten URLs und Fehlermeldungen, bevor sie basierend auf der Synthetics-Konfigurationseinstellung `restrictedUrlParameters` in Protokolle, Berichte, HAR-Dateien und Canary-Lauf-Fehler aufgenommen werden. Sie müssen `getSanitizedUrl` oder `getSanitizedErrorMessage` nur verwenden, wenn Sie URLs oder Fehler in Ihrem Skript protokollieren. Synthetics speichert keine Canaryartefakte mit Ausnahme von Canaryfehlern, die vom Skript ausgelöst werden. Canary-Lauf-Artefakte werden in Ihrem Kundenkonto gespeichert. Weitere Informationen finden Sie unter Sicherheitsüberlegungen für Synthetics-Canaries.

getSanitizedUrl(url, stepConfig = null)

Diese Funktion ist in `syn-nodejs-puppeteer-3.2` und höher verfügbar. Es gibt bereinigt URL-Zeichenfolgen basierend auf der Konfiguration. Sie können sich dafür entscheiden, sensible URL-Parameter wie password und access_token zu schwärzen, indem Sie die Eigenschaft `restrictedUrlParameters` festlegen. Standardmäßig werden Passwörter in URLs geschwärzt. Sie können bei Bedarf URL-Passwörter aktivieren, indem Sie `includeUrlPassword` auf true setzen.

Diese Funktion löst einen Fehler aus, wenn die übergebene URL keine gültige URL ist.

Parameter

- *`url`* ist eine Zeichenfolge und ist die URL, die bereinigt werden soll.
- *`StepConfig`* (Optional) überschreibt die globale Synthetics-Konfiguration für diese Funktion. Wenn `stepConfig` nicht übergeben wird, wird die globale Konfiguration verwendet, um die URL zu bereinigen.

Beispiel

In diesem Beispiel wird die folgende Beispiel-URL verwendet :`https://example.com/learn/home?access_token=12345&token_type=Bearer&expires_in=1200`. In diesem Beispiel enthält `access_token` Ihre vertraulichen Informationen, die nicht protokolliert werden sollten.

Beachten Sie, dass die Synthetics-Services keine Canary-Artefakte speichern. Artefakte wie Protokolle, Screenshots und Berichte werden in einem Amazon-S3-Bucket in Ihrem Kundenkonto gespeichert.

Der erste Schritt besteht darin, die Synthetics-Konfiguration festzulegen.

```
// Import Synthetics dependency
const synthetics = require('Synthetics');

// Import Synthetics logger for logging url
const log = require('SyntheticsLogger');

// Get Synthetics configuration
const synConfig = synthetics.getConfiguration();

// Set restricted parameters
synConfig.setConfig({
   restrictedUrlParameters: ['access_token'];
});
```

Als nächstes bereinigen und protokollieren Sie die URL

```
// Import SyntheticsLogHelper dependency
const syntheticsLogHelper = require('SyntheticsLogHelper');

const sanitizedUrl = synthetics.getSanitizedUrl('https://example.com/learn/home?
access_token=12345&token_type=Bearer&expires_in=1200');
```

Dies protokolliert Folgendes in Ihrem Canaryprotokoll.

```
My example url is: https://example.com/learn/home?
access_token=REDACTED&token_type=Bearer&expires_in=1200
```

Sie können die Synthetics-Konfiguration für eine URL überschreiben, indem Sie einen optionalen Parameter mit Synthetics-Konfigurationsoptionen übergeben, wie im folgenden Beispiel gezeigt.

```
const urlConfig = {
   restrictedUrlParameters = ['*']
};
const sanitizedUrl = synthetics.getSanitizedUrl('https://example.com/learn/home?
access_token=12345&token_type=Bearer&expires_in=1200', urlConfig);
```

```
logger.info('My example url is: ' + sanitizedUrl);
```

Im obigen Beispiel werden alle Abfrageparameter geschwärzt und wie folgt protokolliert:

```
My example url is: https://example.com/learn/home?
access_token=REDACTED&token_type=REDACTED&expires_in=REDACTED
```

getSanitizedErrorNachricht

Diese Funktion ist in `syn-nodejs-puppeteer-3.2` und höher verfügbar. Es gibt bereinigte Fehlerzeichenfolgen zurück, indem alle URLs, die auf der Synthetics-Konfiguration basieren, bereinigt werden. Sie können die globale Synthetics-Konfiguration überschreiben, wenn Sie diese Funktion aufrufen, indem Sie einen optionalen `stepConfig`-Parameter übergeben.

Parameter

- *`error`* ist der zu sanierende Fehler. Es kann ein Fehler-Objekt oder eine Zeichenfolge sein.
- *`StepConfig`* (Optional) überschreibt die globale Synthetics-Konfiguration für diese Funktion. Wenn `stepConfig` nicht übergeben wird, wird die globale Konfiguration verwendet, um die URL zu bereinigen.

Beispiel

In diesem Beispiel wird der folgende Fehler verwendet: `Failed to load url: https://example.com/learn/home?access_token=12345&token_type=Bearer&expires_in=1200`

Der erste Schritt besteht darin, die Synthetics-Konfiguration festzulegen.

```
// Import Synthetics dependency
const synthetics = require('Synthetics');

// Import Synthetics logger for logging url
const log = require('SyntheticsLogger');

// Get Synthetics configuration
const synConfig = synthetics.getConfiguration();

// Set restricted parameters
synConfig.setConfig({
   restrictedUrlParameters: ['access_token'];
```

```
});
```

Als nächstes bereinigen und protokollieren Sie die Fehlermeldung

```
// Import SyntheticsLogHelper dependency
const syntheticsLogHelper = require('SyntheticsLogHelper');

try {
   // Your code which can throw an error containing url which your script logs
} catch (error) {
    const sanitizedErrorMessage = synthetics.getSanitizedErrorMessage(errorMessage);
    logger.info(sanitizedErrorMessage);
}
```

Dies protokolliert Folgendes in Ihrem Canaryprotokoll.

```
Failed to load url: https://example.com/learn/home?
access_token=REDACTED&token_type=Bearer&expires_in=1200
```

getSanitizedHeaders(Header, StepConfig=Null)

Diese Funktion ist in `syn-nodejs-puppeteer-3.2` und höher verfügbar. Es gibt bereinigte Header basierend auf der `restrictedHeaders`-Eigenschaft von `syntheticsConfiguration` zurück. Die in der Eigenschaft `restrictedHeaders` angegebenen Header werden aus Protokollen, HAR-Dateien und Berichten entfernt.

Parameter

- *`Header`* ist ein Objekt, das die zu bereinigenden Header enthält.
- *`StepConfig`* (Optional) überschreibt die globale Synthetics-Konfiguration für diese Funktion. Wenn `stepConfig` nicht übergeben wird, wird die globale Konfiguration verwendet, um die Header zu bereinigen.

Node.js-Bibliotheksklassen und -funktionen, die nur für UI-Canaries gelten

Die folgenden CloudWatch Synthetics-Bibliotheksfunktionen für Node.js sind nur für UI Canaries nützlich.

Themen

- Synthetics-Klasse

- BrokenLinkCheckerReport Klasse
- SyntheticsLink Klasse

Synthetics-Klasse

Die folgenden Funktionen befinden sich in der Klasse Synthetics.

asynchron addUserAgent (Seite,); userAgentString

Diese Funktion hängt *`userAgentString`*an den User-Agent-Header der angegebenen Seite an.

Beispiel:

```
await synthetics.addUserAgent(page, "MyApp-1.0");
```

Bewirkt, dass der User-Agent-Header der Seite auf *`browsers-user-agent-header-value`*`MyApp-1.0` gesetzt wird.

asynchron executeStep (stepName, functionToExecute, [StepConfig]);

Führt den bereitgestellten Schritt aus und bettet ihn in Start/Pass/Fail-Protokollierung, Start/Pass/Fail-Screenshots sowie Pass/Fail- und Dauer-Metriken ein.

> **Note**
>
> Wenn Sie die Laufzeit `syn-nodejs-2.1` oder höher verwenden, können Sie konfigurieren, ob und wann Screenshots erstellt werden. Weitere Informationen finden Sie unter SyntheticsConfiguration Klasse.

Die `executeStep`-Funktion tut Folgendes:

- Protokolliert, dass der Schritt gestartet wurde.
- Erfasst einen Screenshot mit dem Namen `<stepName>-starting`.
- Startet einen Timer.
- Führt die bereitgestellte Funktion aus.
- Wenn die Funktion normal zurückgegeben wird, gilt sie als „bestanden“. Wenn die Funktion scheitert, gilt sie als fehlgeschlagen.

- Beendet den Timer.
- Protokolliert, ob der Schritt bestanden oder fehlgeschlagen ist
- Erfasst einen Screenshot mit dem Namen `<stepName>-succeeded` oder `<stepName>-failed`.
- Gibt die Metrik `stepName SuccessPercent` aus, wobei „100“ für erfolgreich bzw. „0“ für nicht erfolgreich steht.
- Gibt die Metrik `stepName Duration` aus, mit einem Wert basierend auf der Start- und der Endzeit des Schrittes.
- Gibt schließlich zurück, was die `functionToExecute` zurückgegeben hat, oder wiederholt einen Throw-Vorgang von `functionToExecute`.

Wenn der Canary die Laufzeit `syn-nodejs-2.0` oder höher verwendet, fügt diese Funktion dem Bericht des Canary auch eine Zusammenfassung der Schrittausführung hinzu. Die Zusammenfassung enthält Details zu jedem Schritt, wie Startzeit, Endzeit, Status (PASSED/FAILED), Fehlergrund (falls fehlgeschlagen) und Screenshots, die während der Ausführung jedes Schritts erfasst wurden.

Beispiel:

```
await synthetics.executeStep('navigateToUrl', async function (timeoutInMillis = 30000)
 {
           await page.goto(url, {waitUntil: ['load', 'networkidle0'], timeout:
 timeoutInMillis});});
```

Antwort:

Gibt zurück, was `functionToExecute` zurückgibt.

Updates mit syn-nodejs-2.2

Zunächst können Sie optional Schrittkonfigurationen übergeben`syn-nodejs-2.2`, um CloudWatch Synthetics-Konfigurationen auf Schrittebene zu überschreiben. Eine Liste der Optionen, die Sie an `executeStep` übergeben können, finden Sie unter SyntheticsConfiguration Klasse.

Das folgende Beispiel überschreibt die `false`-Standardkonfiguration für `continueOnStepFailure` bis `true` und gibt an, wann Screenshots erstellt werden sollen.

```
var stepConfig = {
    'continueOnStepFailure': true,
```

```
    'screenshotOnStepStart': false,
    'screenshotOnStepSuccess': true,
    'screenshotOnStepFailure': false
}

await executeStep('Navigate to amazon', async function (timeoutInMillis = 30000) {
      await page.goto(url, {waitUntil: ['load', 'networkidle0'], timeout:
 timeoutInMillis});
 }, stepConfig);
```

getDefaultLaunchOptionen ();

Die `getDefaultLaunchOptions()` Funktion gibt die Browser-Startoptionen zurück, die von CloudWatch Synthetics verwendet werden. Weitere Informationen finden Sie unter Startoptionen-Typ

```
// This function returns default launch options used by Synthetics.
const defaultOptions = await synthetics.getDefaultLaunchOptions();
```

getPage();

Gibt die aktuell geöffnete Seite als Puppeteer-Objekt zurück. Weitere Informationen finden Sie unter Puppeteer API v1.14.0.

Beispiel:

```
let page = synthetics.getPage();
```

Antwort:

Die Seite (Puppeteer-Objekt), die derzeit in der aktuellen Browsersitzung geöffnet ist.

getRequestResponseLogHelper();

> **Important**
>
> In Canaries, die die Laufzeit `syn-nodejs-puppeteer-3.2` oder höher verwenden, ist diese Funktion zusammen mit der `RequestResponseLogHelper`-Klasse veraltet. Jede Verwendung dieser Funktion bewirkt, dass eine Warnung in Ihren Canaryprotokollen angezeigt wird. Diese Funktion wird in zukünftigen Laufzeitversionen entfernt. Wenn Sie diese Funktion verwenden, verwenden Sie stattdessen RequestResponseLogHelper Klasse.

Verwenden Sie diese Funktion als Builder-Muster zum Optimieren der Anforderungs- und Antwortprotokollierungsmarkierungen.

Beispiel:

```
synthetics.setRequestResponseLogHelper(getRequestResponseLogHelper().withLogRequestHeaders(fals
```

Antwort:

```
{RequestResponseLogHelper}
```

Start (Optionen)

Die Optionen für diese Funktion sind erst ab der Laufzeit-Version `syn-nodejs-2.1` verfügbar.

Diese Funktion wird nur für UI-Canarys verwendet. Es schließt den vorhandenen Browser und startet einen neuen.

> **Note**
>
> CloudWatch Synthetics startet immer einen Browser, bevor Sie mit der Ausführung Ihres Skripts beginnen. Sie müssen launch() nicht aufrufen, es sei denn, Sie möchten einen neuen Browser mit benutzerdefinierten Optionen starten.

(Optionen) ist ein konfigurierbarer Satz von Optionen, die im Browser festgelegt werden sollen. Weitere Informationen finden Sie unter Startoptionen-Typ.

Wenn Sie diese Funktion ohne Optionen aufrufen, startet Synthetics einen Browser mit Standardargumenten `executablePath` und `defaultViewport`. Das Standardansichtsfenster in CloudWatch Synthetics ist 1920 mal 1080.

Sie können die von CloudWatch Synthetics verwendeten Startparameter überschreiben und beim Starten des Browsers zusätzliche Parameter übergeben. Mit dem folgenden Codeausschnitt wird beispielsweise ein Browser mit Standardargumenten und einem standardmäßigen ausführbaren Pfad gestartet, jedoch mit einem Ansichtsfenster von 800 x 600.

```
await synthetics.launch({
        defaultViewport: {
            "deviceScaleFactor": 1,
```

```
            "width": 800,
            "height": 600
    }});
```

Der folgende Beispielcode fügt den `ignoreHTTPSErrors` CloudWatch Synthetics-Startparametern einen neuen Parameter hinzu:

```
await synthetics.launch({
        ignoreHTTPSErrors: true
 });
```

Sie können die Websicherheit deaktivieren, indem Sie args in den CloudWatch Synthetics-Startparametern ein `--disable-web-security` Flag hinzufügen:

```
// This function adds the --disable-web-security flag to the launch parameters
const defaultOptions = await synthetics.getDefaultLaunchOptions();
const launchArgs = [...defaultOptions.args, '--disable-web-security'];
await synthetics.launch({
     args: launchArgs
  });
```

RequestResponseLogHelper Klasse

> **Important**
>
> In Canaries, die die Laufzeit `syn-nodejs-puppeteer-3.2` oder höher verwenden, ist diese Klasse veraltet. Jede Verwendung dieser Klasse bewirkt, dass eine Warnung in Ihren Canaryprotokollen angezeigt wird. Diese Funktion wird in zukünftigen Laufzeitversionen entfernt. Wenn Sie diese Funktion verwenden, verwenden Sie stattdessen RequestResponseLogHelper Klasse.

Behandelt die präzise Konfiguration und Erstellung von Zeichenfolgendarstellungen von Anforderungs- und Antwort-Nutzlasten.

```
class RequestResponseLogHelper {

    constructor () {
        this.request = {url: true, resourceType: false, method: false, headers: false,
 postData: false};
```

```
        this.response = {status: true, statusText: true, url: true, remoteAddress:
 false, headers: false};
    }

    withLogRequestUrl(logRequestUrl);

    withLogRequestResourceType(logRequestResourceType);

    withLogRequestMethod(logRequestMethod);

    withLogRequestHeaders(logRequestHeaders);

    withLogRequestPostData(logRequestPostData);


    withLogResponseStatus(logResponseStatus);

    withLogResponseStatusText(logResponseStatusText);

    withLogResponseUrl(logResponseUrl);

    withLogResponseRemoteAddress(logResponseRemoteAddress);

    withLogResponseHeaders(logResponseHeaders);
```

Beispiel:

```
synthetics.setRequestResponseLogHelper(getRequestResponseLogHelper()
.withLogRequestPostData(true)
.withLogRequestHeaders(true)
.withLogResponseHeaders(true));
```

Antwort:

```
{RequestResponseLogHelper}
```

setRequestResponseLogHelper();

> **Important**
>
> In Canaries, die die Laufzeit `syn-nodejs-puppeteer-3.2` oder höher verwenden, ist diese Funktion zusammen mit der `RequestResponseLogHelper`-Klasse veraltet.

Jede Verwendung dieser Funktion bewirkt, dass eine Warnung in Ihren Canaryprotokollen angezeigt wird. Diese Funktion wird in zukünftigen Laufzeitversionen entfernt. Wenn Sie diese Funktion verwenden, verwenden Sie stattdessen RequestResponseLogHelper Klasse.

Verwenden Sie diese Funktion als Builder-Muster zum Einstellen der Anforderungs- und Antwortprotokollierungsmarkierungen.

Beispiel:

```
synthetics.setRequestResponseLogHelper().withLogRequestHeaders(true).withLogResponseHeaders(tru
```

Antwort:

```
{RequestResponseLogHelper}
```

async takeScreenshot(name, suffix);

Erstelle einen Screenshot (.PNG) der aktuellen Seite mit Name und Suffix (Optional).

Beispiel:

```
await synthetics.takeScreenshot("navigateToUrl", "loaded")
```

In diesem Beispiel wird ein Screenshot mit dem Namen `01-navigateToUrl-loaded.png` aufgenommen und in den S3 Bucket des Canary hochgeladen.

Sie können einen Screenshot für einen bestimmten Canary-Schritt erstellen, indem Sie das `stepName` als ersten Parameter übergeben. Screenshots sind mit dem Canaryschritt in Ihren Berichten verknüpft, damit Sie jeden Schritt beim Debuggen verfolgen können.

CloudWatch Synthetics Canaries macht automatisch Screenshots vor dem Starten eines Schritts (die `executeStep` Funktion) und nach Abschluss des Schritts (es sei denn, Sie konfigurieren den Canary so, dass Screenshots deaktiviert werden). Sie können weitere Screenshots erstellen, indem Sie den Schrittnamen in der `takeScreenshot`-Funktion übergeben.

Im folgenden Beispiel wird ein Screenshot mit dem `signupForm` als Wert von `stepName` erstellt. Der Screenshot erhält den Namen `02-signupForm-address` und wird mit dem Schritt namens `signupForm` im Canary-Bericht verknüpft.

```
await synthetics.takeScreenshot('signupForm', 'address')
```

BrokenLinkCheckerReport Klasse

Diese Klasse bietet Methoden, um eine Synthetics-Verknüpfung hinzuzufügen. Es wird nur auf Canarysn unterstützt, die die `syn-nodejs-2.0-beta`-Version der Laufzeitumgebung oder höher verwenden.

Um `BrokenLinkCheckerReport` zu verwenden, fügen Sie die folgenden Zeilen in das Skript ein:

```
const BrokenLinkCheckerReport = require('BrokenLinkCheckerReport');

const brokenLinkCheckerReport = new BrokenLinkCheckerReport();
```

Nützliche Funktionsdefinitionen:

addLink(***`syntheticsLink`***, isBroken)

`syntheticsLink` ist ein `SyntheticsLink`-Objekt, das eine Verknüpfung darstellt. Diese Funktion fügt den Link entsprechend dem Statuscode hinzu. Standardmäßig betrachtet es einen Link als unterbrochen, wenn der Statuscode nicht verfügbar ist oder der Statuscode 400 oder höher ist. Sie können dieses Standardverhalten überschreiben, indem Sie den optionalen Parameter `isBrokenLink` mit einem Wert von`true` oder `false` übergeben.

Diese Funktion hat keinen Rückgabewert.

getLinks()

Diese Funktion gibt ein Array von `SyntheticsLink`-Objekten zurück, die im Bericht zur Überprüfung von defekten Links enthalten sind.

getTotalBrokenLinks ()

Diese Funktion gibt eine Zahl zurück, die die Gesamtzahl der defekten Links darstellt.

getTotalLinksGeprüft ()

Diese Funktion gibt eine Zahl zurück, die die Gesamtzahl der im Bericht enthaltenen Links darstellt.

Wie benutzt man BrokenLinkCheckerReport

Das folgende Code-Snippet aus einem Canary-Skript veranschaulicht anhand eines Beispiels die Navigation zu einem Link und das Hinzufügen zum Bericht zur Überprüfung eines fehlerhaften Links.

1. Importieren von `SyntheticsLink`, `BrokenLinkCheckerReport` und `Synthetics`.

```
const BrokenLinkCheckerReport = require('BrokenLinkCheckerReport');
const SyntheticsLink = require('SyntheticsLink');

// Synthetics dependency
const synthetics = require('Synthetics');
```

2. Um einen Link zum Bericht hinzuzufügen, erstellen Sie eine Instance von `BrokenLinkCheckerReport`.

```
let brokenLinkCheckerReport = new BrokenLinkCheckerReport();
```

3. Navigieren Sie zu der URL, und fügen Sie sie dem Bericht zur Überprüfung fehlerhafter Links hinzu.

```
let url = "https://amazon.com";

let syntheticsLink = new SyntheticsLink(url);

// Navigate to the url.
let page = await synthetics.getPage();

// Create a new instance of Synthetics Link
let link = new SyntheticsLink(url)

try {
    const response = await page.goto(url, {waitUntil: 'domcontentloaded', timeout:
 30000});
} catch (ex) {
    // Add failure reason if navigation fails.
    link.withFailureReason(ex);
}

if (response) {
    // Capture screenshot of destination page
    let screenshotResult = await synthetics.takeScreenshot('amazon-home', 'loaded');

    // Add screenshot result to synthetics link
    link.addScreenshotResult(screenshotResult);

    // Add status code and status description to the link
    link.withStatusCode(response.status()).withStatusText(response.statusText())
```

```
}

// Add link to broken link checker report.
brokenLinkCheckerReport.addLink(link);
```

4. Fügen Sie den Bericht zu Synthetics hinzu. Dadurch wird für jeden Canary-Lauf eine JSON-Datei mit dem Namen `BrokenLinkCheckerReport.json` in Ihrem S3 Bucket erstellt. Sie können einen Link-Bericht in der Konsole für jeden Canary-Lauf zusammen mit Screenshots, Protokollen und HAR-Dateien sehen.

```
await synthetics.addReport(brokenLinkCheckerReport);
```

SyntheticsLink Klasse

Diese Klasse bietet Methoden zum Umschließen von Informationen. Es wird nur auf Canaries unterstützt, die die `syn-nodejs-2.0-beta`-Version der Laufzeit oder höher verwenden.

Um `SyntheticsLink` zu verwenden, fügen Sie die folgenden Zeilen in das Skript ein:

```
const SyntheticsLink = require('SyntheticsLink');

const syntheticsLink = new SyntheticsLink("https://www.amazon.com");
```

Diese Funktion gibt `syntheticsLink`*`Object`* zurück

Nützliche Funktionsdefinitionen:

withUrl(***`url`***)

`url` ist eine URL-Zeichenfolge. Diese Funktion gibt `syntheticsLink`*`Object`* zurück

withText(***`text`***)

`text` ist eine Zeichenfolge, die Ankertext darstellt. Diese Funktion gibt `syntheticsLink`*`Object`* zurück. Es fügt Ankertext hinzu, der dem Link entspricht.

withParentUrl(***`Eltern-URL`***)

`parentUrl` ist eine Zeichenfolge, die die übergeordnete URL (Quellseite) darstellt. Diese Funktion gibt `syntheticsLink`*`Object`* zurück

withStatusCode(*`statusCode`*)

`statusCode` ist eine Zeichenfolge, die den Statuscode darstellt. Diese Funktion gibt `syntheticsLink`*`Object`* zurück

withFailureReason***`(Grund des Fehlers)`***

`failureReason` ist eine Zeichenfolge, die den Grund für den Fehler darstellt. Diese Funktion gibt `syntheticsLink`*`Object`* zurück

addScreenshotResult(***`Ergebnis des Screenshots`***)

`screenshotResult` ist ein Objekt. Es ist eine Instance von `ScreenshotResult`, die von der Synthetics-Funktion `takeScreenshot` zurückgegeben wurde. Das Objekt umfasst Folgendes:

- `fileName` – Eine Zeichenfolge, die `screenshotFileName` darstellt
- `pageUrl` (optional)
- `error` (optional)

Node.js-Bibliotheksklassen und -funktionen, die nur für API-Canaries gelten

Die folgenden CloudWatch Synthetics-Bibliotheksfunktionen für Node.js sind nur für API Canaries nützlich.

Themen

executeHttpStep(stepName, RequestOptions, [Rückruf], [StepConfig])

Führt die bereitgestellte HTTP-Anforderung als Schritt aus und veröffentlicht `SuccessPercent`-(pass/fail) und `Duration`-Metriken.

executeHttpStepverwendet entweder native HTTP- oder HTTPS-Funktionen unter der Haube, abhängig vom in der Anfrage angegebenen Protokoll.

Diese Funktion fügt dem Bericht des Canarys außerdem eine Zusammenfassung der Schrittausführung hinzu. Die Zusammenfassung enthält Details zu jeder HTTP-Anforderung, wie zum Beispiel die folgenden:

- Startzeit
- Endzeit
- Status (PASSED/FAILED)
- Fehlergrund, wenn fehlgeschlagen
- HTTP-Aufrufdetails wie Anfrage/Antwort-Header, Text, Statuscode, Statusmeldung und Leistungs-Timings.

Parameter

stepName(***String***)

Legt den Namen des Schrittes fest. Dieser Name wird auch für die Veröffentlichung von CloudWatch Metriken für diesen Schritt verwendet.

requestOptions(***Object or String***)

Der Wert dieses Parameters kann eine URL, eine URL-Zeichenfolge oder ein Objekt sein. Wenn es sich um ein Objekt handelt, muss es sich um eine Gruppe konfigurierbarer Optionen handeln, um eine HTTP-Anforderung zu stellen. Es unterstützt alle Optionen in http.request(options[, callback]) in der Node.js-Dokumentation.

Zusätzlich zu diesen Node.js-Optionen unterstützt requestOptions den zusätzlichen Parameter `body`. Sie können den Parameter `body` verwenden, um Daten als Anforderungstext zu übergeben.

callback(***response***)

(Optional) Dies ist eine Benutzerfunktion, die mit der HTTP-Antwort aufgerufen wird. Die Antwort ist vom Typ Class: http. IncomingMessage.

stepConfig(***object***)

(Optional) Verwenden Sie diesen Parameter, um globale Synthetics-Konfigurationen mit einer anderen Konfiguration für diesen Schritt zu überschreiben.

Beispiele für die Verwendung executeHttpStep

Die folgende Reihe von Beispielen baut aufeinander auf, um die verschiedenen Verwendungsmöglichkeiten dieser Option zu veranschaulichen.

In diesem ersten Beispiel werden Anforderungsparameter konfiguriert. Sie können eine URL als RequestOptions übergeben:

```
let requestOptions = 'https://www.amazon.com';
```

Oder Sie können eine Reihe von Optionen übergeben:

```
let requestOptions = {
        'hostname': 'myproductsEndpoint.com',
        'method': 'GET',
        'path': '/test/product/validProductName',
        'port': 443,
        'protocol': 'https:'
    };
```

Das nächste Beispiel erstellt eine Callback-Funktion, die eine Antwort akzeptiert. Wenn Sie keinen Callback angeben, überprüft CloudWatch Synthetics standardmäßig, ob der Status zwischen 200 und 299 (einschließlich) liegt.

```
// Handle validation for positive scenario
    const callback = async function(res) {
        return new Promise((resolve, reject) => {
            if (res.statusCode < 200 || res.statusCode > 299) {
                throw res.statusCode + ' ' + res.statusMessage;
            }

            let responseBody = '';
            res.on('data', (d) => {
                responseBody += d;
            });

            res.on('end', () => {
                // Add validation on 'responseBody' here if required. For ex, your
 status code is 200 but data might be empty
                resolve();
            });
        });
    };
```

Das nächste Beispiel erstellt eine Konfiguration für diesen Schritt, die die globale CloudWatch Synthetics-Konfiguration überschreibt. Die Schrittkonfiguration in diesem Beispiel ermöglicht

Anforderungskopfzeilen, Antwort-Header, Anforderungstext (Postdaten) und Antworttext in Ihrem Bericht und schränkt die Header-Werte „X-AMZ-Security-Token“ und „Authorization“ ein. Standardmäßig sind diese Werte aus Sicherheitsgründen nicht im Bericht enthalten. Wenn Sie sie einbeziehen, werden die Daten nur in Ihrem S3 Bucket gespeichert.

```
// By default headers, post data, and response body are not included in the report for
 security reasons.
// Change the configuration at global level or add as step configuration for individual
 steps
let stepConfig = {
    includeRequestHeaders: true,
    includeResponseHeaders: true,
    restrictedHeaders: ['X-Amz-Security-Token', 'Authorization'], // Restricted header
 values do not appear in report generated.
    includeRequestBody: true,
    includeResponseBody: true
};
```

Dieses letzte Beispiel leitet Ihre Anfrage an den Schritt weiter executeHttpStepund benennt ihn.

```
await synthetics.executeHttpStep('Verify GET products API', requestOptions, callback,
 stepConfig);
```

Mit diesen Beispielen fügt CloudWatch Synthetics die Details aus jedem Schritt in Ihrem Bericht hinzu und erstellt mithilfe von stepName Metriken für jeden Schritt.

Sie sehen `successPercent`- und `duration`-Metriken für den Schritt `Verify GET products API`. Sie können Ihre API-Leistung überwachen, indem Sie die Metriken für Ihre API-Aufrufschritte überwachen.

Ein vollständiges Beispielskript, das diese Funktionen verwendet, finden Sie unter Mehrstufiger API-Canary.

Verfügbare Bibliotheksfunktionen für Python-Canary-Skripte mit Selenium

Dieser Abschnitt listet die Selenium-Bibliotheksfunktionen auf, die für Python-Canary-Skripte verfügbar sind

Themen

- Python- und Selenium-Bibliotheksklassen und Funktionen, die für alle Canarys gelten
- Python- und Selenium-Bibliotheksklassen und Funktionen, die nur für UI-Canarys gelten

Python- und Selenium-Bibliotheksklassen und Funktionen, die für alle Canarys gelten

Die folgenden CloudWatch Synthetics Selenium-Bibliotheksfunktionen für Python sind für alle Kanarienvögel nützlich.

Themen

- SyntheticsConfiguration Klasse
- SyntheticsLogger Klasse

SyntheticsConfiguration Klasse

Sie können die SyntheticsConfiguration Klasse verwenden, um das Verhalten von Synthetics-Bibliotheksfunktionen zu konfigurieren. Sie können diese Klasse beispielsweise verwenden, um die `executeStep()`-Funktion so zu konfigurieren, dass keine Screenshots erfasst werden.

Sie können CloudWatch Synthetics-Konfigurationen auf globaler Ebene festlegen.

Funktionsdefinitionen:

set_config (Optionen)

```
from aws_synthetics.common import synthetics_configuration
```

`options` ist ein Objekt, bei dem es sich um eine Reihe konfigurierbarer Optionen für Ihren Canary handelt. In den folgenden Abschnitten werden die möglichen Felder in *`options`* erläutert.

- `screenshot_on_step_start` (boolean) – Gibt an, ob ein Screenshot erstellt werden soll, bevor ein Schritt gestartet wird.
- `screenshot_on_step_success` (boolean) – Gibt an, ob nach einem erfolgreichen Schritt ein Screenshot erstellt werden soll.
- `screenshot_on_step_failure` (boolean) - Ob ein Screenshot erstellt werden soll, nachdem ein Schritt fehlgeschlagen ist.

with_screenshot_on_step_start(screenshot_on_step_start)

Akzeptiert ein boolesches Argument, das angibt, ob ein Screenshot erstellt werden soll, bevor ein Schritt gestartet wird.

with_screenshot_on_step_success(screenshot_on_step_success)

Akzeptiert ein boolesches Argument, das angibt, ob nach erfolgreichem Abschluss eines Schritts ein Screenshot erstellt werden soll.

with_screenshot_on_step_failure(screenshot_on_step_failure)

Akzeptiert ein boolesches Argument, das angibt, ob nach einem Schritt ein Screenshot erstellt werden soll.

get_screenshot_on_step_start()

Gibt zurück, ob ein Screenshot erstellt werden soll, bevor ein Schritt gestartet wird.

get_screenshot_on_step_success()

Gibt zurück, ob ein Screenshot erstellt werden soll, nachdem ein Schritt erfolgreich abgeschlossen wurde.

get_screenshot_on_step_failure()

Gibt zurück, ob ein Screenshot erstellt werden soll, nachdem ein Schritt fehlgeschlagen ist.

disable_step_screenshots()

Deaktiviert alle Screenshot-Optionen (get_screenshot_on_step_start, get_screenshot_on_step_success und get_screenshot_on_step_failure).

enable_step_screenshots()

Aktiviert alle Screenshot-Optionen (get_screenshot_on_step_start, get_screenshot_on_step_success und get_screenshot_on_step_failure). Standardmäßig sind alle diese Methoden aktiviert.

SetConfig (Optionen) in Bezug auf Metriken CloudWatch

Für Canarys, die `syn-python-selenium-1.1` oder höher verwenden, können die (Optionen) für setConfig die folgenden booleschen Parameter enthalten, die bestimmen, welche Metriken vom Canary veröffentlicht werden. Der Standardwert für jede dieser Optionen ist `true`. Die Optionen, die mit `aggregated` beginnen, bestimmen, ob die Metrik ohne die `CanaryName`-Dimension ausgegeben wird. Sie können diese Metriken verwenden, um die aggregierten Ergebnisse für alle Canarys anzuzeigen. Die anderen Optionen bestimmen, ob die Metrik mit der `CanaryName` Dimension

ausgegeben wird. Sie können diese Metriken verwenden, um die Ergebnisse für jeden einzelnen Canary anzuzeigen.

Eine Liste der von Canaries ausgegebenen CloudWatch Messwerte finden Sie unter. CloudWatch von Canaries veröffentlichte Metriken

- `failed_canary_metric` (boolean) – Gibt an, ob die `Failed`-Metrik (mit der `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `failed_requests_metric` (boolean) – Gibt an, ob die `Failed requests`-Metrik (mit der `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `2xx_metric` (boolean) – Gibt an, ob die `2xx`-Metrik (mit der `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `4xx_metric` (boolean) – Gibt an, ob die `4xx`-Metrik (mit der `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `5xx_metric` (boolean) – Gibt an, ob die `5xx`-Metrik (mit der `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `step_duration_metric` (boolean) – Gibt an, ob die `Step duration`-Metrik (mit der `CanaryName`-Dimension `StepName`) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `step_success_metric` (boolean) – Gibt an, ob die `Step success`-Metrik (mit der `CanaryName`-Dimension `StepName`) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `aggregated_failed_canary_metric` (boolean) – Gibt an, ob die `Failed`-Metrik (ohne die `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `aggregated_failed_requests_metric` (boolean) – Gibt an, ob die `Failed Requests`-Metrik (ohne die `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `aggregated_2xx_metric` (boolean) – Gibt an, ob die `2xx`-Metrik (ohne die `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `aggregated_4xx_metric` (boolean) – Gibt an, ob die `4xx`-Metrik (ohne die `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.
- `aggregated_5xx_metric` (boolean) – Gibt an, ob die `5xx`-Metrik (ohne die `CanaryName`-Dimension) für diesen Canary emittiert werden soll. Der Standardwert ist `true`.

with_2xx_metric(2xx_metric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `2xx`-Metrik mit der `CanaryName`-Dimension für diesen Canary emittiert werden soll.

with_4xx_metric(4xx_metric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `4xx`-Metrik mit der `CanaryName`-Dimension für diesen Canary emittiert werden soll.

with_5xx_metric(5xx_metric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `5xx`-Metrik mit der `CanaryName`-Dimension für diesen Canary emittiert werden soll.

withAggregated2xxMetric(aggregated2xxMetric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `2xx`-Metrik mit keiner Dimension für diesen Canary emittiert werden soll.

withAggregated4xxMetric(aggregated4xxMetric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `4xx`-Metrik mit keiner Dimension für diesen Canary emittiert werden soll.

with_aggregated_5xx_metric(aggregated_5xx_metric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `5xx`-Metrik mit keiner Dimension für diesen Canary emittiert werden soll.

with_aggregated_failed_canary_metric(aggregated_failed_canary_metric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `Failed`-Metrik mit keiner Dimension für diesen Canary emittiert werden soll.

with_aggregated_failed_requests_metric(aggregated_failed_requests_metric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `Failed requests`-Metrik mit keiner Dimension für diesen Canary emittiert werden soll.

with_failed_canary_metric(failed_canary_metric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `Failed`-Metrik mit der `CanaryName`-Dimension für diesen Canary emittiert werden soll.

with_failed_requests_metric(failed_requests_metric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `Failed requests`-Metrik mit der `CanaryName`-Dimension für diesen Canary emittiert werden soll.

with_step_duration_metric(step_duration_metric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `Duration`-Metrik mit der `CanaryName`-Dimension für diesen Canary emittiert werden soll.

with_step_success_metric(step_success_metric)

Akzeptiert ein boolesches Argument, das angibt, ob eine `StepSuccess`-Metrik mit der `CanaryName`-Dimension für diesen Canary emittiert werden soll.

Methoden zum Aktivieren oder Deaktivieren von Metriken

disable_aggregated_request_metrics()

Verhindert, dass der Canary alle Anforderungsmesswerte ausgibt, die ohne `CanaryName`-Dimension ausgegeben werden.

disable_request_metrics()

Deaktiviert alle Anforderungsmetriken, einschließlich aller Canary-Metriken und Metriken, die über alle Canarys aggregiert werden.

disable_step_metrics()

Deaktiviert alle Schrittmetriken, einschließlich Metriken für den Schritterfolg und für die Schrittdauer.

enable_aggregated_request_metrics()

Ermöglicht dem Canary, alle Anforderungsmesswerte auszugeben, die ohne `CanaryName`-Dimension ausgegeben werden.

enable_request_metrics()

Aktiviert alle Anforderungsmetriken, einschließlich aller Canary-Metriken und Metriken, die über alle Canarys aggregiert werden.

enable_step_metrics()

Aktiviert alle Schrittmetriken, einschließlich Metriken für den Schritterfolg und für die Schrittdauer.

Verwendung in UI-Canarys

Importieren Sie zuerst die Synthetics-Abhängigkeit und holen Sie die Konfiguration ab. Legen Sie dann die Konfiguration für jede Option fest, indem Sie die setConfig-Methode mit einer der folgenden Optionen aufrufen.

```
from aws_synthetics.common import synthetics_configuration

synthetics_configuration.set_config(
     {
        "screenshot_on_step_start": False,
        "screenshot_on_step_success": False,
        "screenshot_on_step_failure": True
     }
)

or
```

Oder

```
synthetics_configuration.with_screenshot_on_step_start(False).with_screenshot_on_step_success(F
```

Um alle Screenshots zu deaktivieren, verwenden Sie die Funktion disableStepScreenshots () wie in diesem Beispiel.

```
synthetics_configuration.disable_step_screenshots()
```

Sie können Screenshots jederzeit im Code aktivieren oder deaktivieren. Wenn Sie beispielsweise Screenshots nur für einen Schritt deaktivieren möchten, deaktivieren Sie sie, bevor Sie diesen Schritt ausführen, und aktivieren Sie sie dann nach dem Schritt.

set_config (Optionen) für UI-Canarys

Ab `syn-python-selenium-1.1` kann `set_config` für UI-Canarys folgende boolesche Parameter enthalten:

- `continue_on_step_failure` (boolean) - Gibt an, b das Canary-Skript nach einem fehlgeschlagenen Schritt weiter ausgeführt werden soll (dies bezieht sich auf die Funktion

executeStep). Wenn Schritte fehlschlagen, wird der Canary-Lauf weiterhin als fehlgeschlagen markiert. Der Standardwert ist `false`.

SyntheticsLogger Klasse

`synthetics_logger` schreibt Protokolle auf die Konsole und in eine lokale Protokolldatei auf derselben Protokollebene. Diese Protokolldatei wird nur dann an beide Speicherorte geschrieben, wenn die Protokollebene auf oder unter der gewünschten Protokollierungsebene der aufgerufenen Protokollfunktion liegt.

Den Protokollierungsanweisungen in der lokalen Protokolldatei werden je nach der Protokollebene der aufgerufenen Funktion „DEBUG“ „INFO“ usw. vorangestellt.

Die Verwendung von `synthetics_logger` ist nicht erforderlich, um eine Protokolldatei zu erstellen, die in Ihren Amazon-S3-Ergebnisspeicherort hochgeladen wird. Sie können stattdessen eine andere Protokolldatei im `/tmp`-Ordner erstellen. Alle Dateien, die unter dem `/tmp`-Ordner erstellt wurden, werden als Artefakte an den Ergebnisspeicherort im S3 Bucket hochgeladen.

Verwendung von `synthetics_logger`:

```
from aws_synthetics.common import synthetics_logger
```

Nützliche Funktionsdefinitionen:

Protokollebene abrufen:

```
log_level = synthetics_logger.get_level()
```

Protokollebene festlegen:

```
synthetics_logger.set_level()
```

Protokollieren Sie eine Nachricht mit einer angegebenen Ebene. Die Ebene kann `DEBUG`, `INFO`, `WARN` oder `ERROR` sein, wie in den folgenden Syntaxbeispielen:

```
synthetics_logger.debug(message, *args, **kwargs)
```

```
synthetics_logger.info(message, *args, **kwargs)
```

```
synthetics_logger.log(message, *args, **kwargs)
```

```
synthetics_logger.warn(message, *args, **kwargs)
```

```
synthetics_logger.error(message, *args, **kwargs)
```

Informationen zu Debug-Parametern finden Sie in der Python-Standarddokumentation unter logging.debug

In diesen Protokollierungsfunktionen ist das `message` die Zeichenfolge für das Nachrichtenformat. Die `args` sind die Argumente, die mit dem Zeichenfolgen-Formatierungsoperator in `msg` zusammengeführt werden.

Es gibt drei Schlüsselwortargumente in `kwargs`:

- `exc_info` – Wird nicht als false ausgewertet, fügt der Protokollierungsmeldung Ausnahmeinformationen hinzu.
- `stack_info` – Standardwert „false“. Bei „true“ werden der Protokollierungsnachricht Stackinformationen hinzugefügt, einschließlich des eigentlichen Protokollierungsaufrufs.
- `extra` – Das dritte optionale Schlüsselwortargument, mit dem Sie ein Wörterbuch übergeben können, das verwendet wird, um das `__dict__` von `LogRecord`, das für das Protokollierungsereignis erstellt wurde, mit benutzerdefinierten Attributen aufzufüllen.

Beispiele:

Protokollieren Sie eine Nachricht mit der Ebene `DEBUG`:

```
synthetics_logger.debug('Starting step - login.')
```

Protokollieren Sie eine Nachricht mit der Ebene `INFO`. `logger.log` ist ein Synonym für `logger.info`:

```
synthetics_logger.info('Successfully completed step - login.')
```

or

```
synthetics_logger.log('Successfully completed step - login.')
```

Protokollieren Sie eine Nachricht mit der Ebene `WARN`:

```
synthetics_logger.warn('Warning encountered trying to publish %s', 'CloudWatch Metric')
```

Protokollieren Sie eine Nachricht mit der Ebene `ERROR`:

```
synthetics_logger.error('Error encountered trying to publish %s', 'CloudWatch Metric')
```

Protokollieren einer Ausnahme:

```
synthetics_logger.exception(message, *args, **kwargs)
```

Protokolliert eine Nachricht mit der Ebene `ERROR`. Ausnahmeinformationen werden der Protokollierungsmeldung hinzugefügt. Sie sollten diese Funktion nur von einem Exception-Handler aufrufen.

Informationen zu Ausnahmeparametern finden Sie in der Standard-Python-Dokumentation unter Logging.Exception.

Das Format für die Nachricht ist `message`. Die `args` sind die Argumente, die mit dem Zeichenfolgen-Formatierungsoperator in `msg` zusammengeführt werden.

Es gibt drei Schlüsselwortargumente in `kwargs`:

- `exc_info` – Wird nicht als false ausgewertet, fügt der Protokollierungsmeldung Ausnahmeinformationen hinzu.
- `stack_info` – Standardwert „false“. Bei „true“ werden der Protokollierungsnachricht Stackinformationen hinzugefügt, einschließlich des eigentlichen Protokollierungsaufrufs.
- `extra` – Das dritte optionale Schlüsselwortargument, mit dem Sie ein Wörterbuch übergeben können, das verwendet wird, um das `__dict__` von `LogRecord`, das für das Protokollierungsereignis erstellt wurde, mit benutzerdefinierten Attributen aufzufüllen.

Beispiel:

```
synthetics_logger.exception('Error encountered trying to publish %s', 'CloudWatch
 Metric')
```

Python- und Selenium-Bibliotheksklassen und Funktionen, die nur für UI-Canarys gelten

Die folgenden CloudWatch Synthetics Selenium-Bibliotheksfunktionen für Python sind nur für UI Canaries nützlich.

Themen

SyntheticsBrowser Klasse

Wenn Sie eine Browserinstance durch Aufrufen von `synthetics_webdriver.Chrome()` erstellen, ist die zurückgegebene Browserinstance vom Typ `SyntheticsBrowser`. Die `SyntheticsBrowser` Klasse steuert den Browser und ermöglicht es dem Canary-Skript ChromeDriver, den Browser zu steuern, sodass Selenium WebDriver mit Synthetics arbeiten kann.

Zusätzlich zu den standardmäßigen Selenium-Methoden bietet es auch die folgenden Methoden.

set_viewport_size (Breite, Höhe)

Legt das Ansichtsfenster des Browsers fest. Beispiel:

```
browser.set_viewport_size(1920, 1080)
```

save_screenshot (Dateiname, Suffix)

Speichert Screenshots im `/tmp`-Verzeichnis. Die Screenshots werden von dort in den Ordner Canary-Artefakte im S3 Bucket hochgeladen.

filename ist der Dateiname für den Screenshot und suffix ist eine optionale Zeichenfolge, die zum Benennen des Screenshots verwendet wird.

Beispiel:

```
browser.save_screenshot('loaded.png', 'page1')
```

SyntheticsWebDriver Klasse

Um diese Klasse zu verwenden, verwenden Sie Folgendes in Ihrem Skript:

```
from aws_synthetics.selenium import synthetics_webdriver
```

add_execution_error (errorMessage, ex);

`errorMessage` beschreibt den Fehler und `ex` ist die aufgetretene Ausnahme

Sie können Folgendes verwenden:`add_execution_error`, um Ausführungsfehler für Ihren Canary festzulegen. Es lässt den Canary fehlschlagen, ohne die Skriptausführung zu unterbrechen. Es wirkt sich auch nicht auf Ihre `successPercent`-Metriken aus.

Sie sollten Fehler nur dann als Ausführungsfehler verfolgen, wenn sie nicht wichtig sind, um den Erfolg oder Misserfolg Ihres Canary-Skripts anzuzeigen.

Ein Beispiel für die Verwendung von `add_execution_error` ist das folgende. Sie überwachen die Verfügbarkeit Ihres Endpunkts und machen Screenshots, nachdem die Seite geladen wurde. Da der Fehler beim Erstellen eines Screenshots die Verfügbarkeit des Endpunkts nicht bestimmt, können Sie beim Erstellen von Screenshots aufgetretene Fehler abfangen und sie als Ausführungsfehler hinzufügen. Ihre Verfügbarkeitsmetriken zeigen weiterhin an, dass der Endpunkt aktiv ist und ausgeführt wird, aber Ihr Canary-Status wird als fehlgeschlagen markiert. Der folgende Codeblock fängt einen solchen Fehler ab und fügt ihn als Ausführungsfehler hinzu.

```
try:
    browser.save_screenshot("loaded.png")
except Exception as ex:
   self.add_execution_error("Unable to take screenshot", ex)
```

add_user_agent (user_agent_str)

Hängt den Wert von `user_agent_str` an den Benutzer-Agent-Header des Browsers an. Sie müssen `user_agent_str` zuweisen, bevor Sie die Browserinstance erstellen.

Beispiel:

```
synthetics_webdriver.add_user_agent('MyApp-1.0')
```

execute_step (step_name, function_to_execute)

Verarbeitet eine Funktion. Folgendes wird ebenfalls bewirkt:

- Protokolliert, dass der Schritt gestartet wurde.
- Erfasst einen Screenshot mit dem Namen `<stepName>-starting`.
- Startet einen Timer.

- Führt die bereitgestellte Funktion aus.
- Wenn die Funktion normal zurückgegeben wird, gilt sie als „bestanden“. Wenn die Funktion scheitert, gilt sie als fehlgeschlagen.
- Beendet den Timer.
- Protokolliert, ob der Schritt bestanden oder fehlgeschlagen ist
- Erfasst einen Screenshot mit dem Namen `<stepName>-succeeded` oder `<stepName>-failed`.
- Gibt die Metrik `stepName` `SuccessPercent` aus, wobei „100“ für erfolgreich bzw. „0“ für nicht erfolgreich steht.
- Gibt die Metrik `stepName` `Duration` aus, mit einem Wert basierend auf der Start- und der Endzeit des Schrittes.
- Gibt schließlich zurück, was die `functionToExecute` zurückgegeben hat, oder wiederholt einen Throw-Vorgang von `functionToExecute`.

Beispiel:

```
from selenium.webdriver.common.by import By

def custom_actions():
        #verify contains
        browser.find_element(By.XPATH, "//*[@id=\"id_1\"][contains(text(),'login')]")
        #click a button
        browser.find_element(By.XPATH, '//*[@id="submit"]/a').click()

    await synthetics_webdriver.execute_step("verify_click", custom_actions)
```

Chrome()

Startet eine Instance des Chromium-Browsers und gibt die erstellte Instance des Browsers zurück.

Beispiel:

```
browser = synthetics_webdriver.Chrome()
browser.get("https://example.com/)
```

Verwenden Sie folgende Parameter, um einen Browser im Inkognito-Modus zu starten:

```
add_argument('—-incognito')
```

Verwenden Sie folgende Parameter, um Proxy-Einstellungen hinzuzufügen:

```
add_argument('--proxy-server=%s' % PROXY)
```

Beispiel:

```
from selenium.webdriver.chrome.options import Options
chrome_options = Options()
chrome_options.add_argument("—-incognito")
browser = syn_webdriver.Chrome(chrome_options=chrome_options)
```

Planen von Canary-Durchläufen mit Cron

Die Verwendung eines Cron-Ausdrucks gibt Ihnen Flexibilität, wenn Sie einen Canary planen. Cron-Ausdrücke enthalten fünf oder sechs Felder in der in der folgenden Tabelle aufgeführten Reihenfolge. Die Felder werden durch Leerzeichen voneinander getrennt. Die Syntax unterscheidet sich je nachdem, ob Sie die CloudWatch Konsole zum Erstellen des Canary verwenden, AWS CLI oder AWS die SDKs. Wenn Sie die Konsole verwenden, geben Sie nur die ersten fünf Felder an. Wenn Sie die AWS SDKs AWS CLI oder verwenden, geben Sie alle sechs Felder an, und Sie müssen * für das `Year` Feld angeben.

Feld	Zulässige Werte	Zulässige Sonderzeichen
Minuten	0-59	, - * /
Stunden	0-23	, - * /
D ay-of-month	1-31	, - * ? / L W
Monat	1-12 oder JAN-DEC	, - * /
D ay-of-week	1-7 oder SUN-SAT	, - * ? / L #
Jahr	*	

Sonderzeichen

- Das , (Komma) enthält mehrere Werte im Ausdruck für ein Feld. Im Feld Monat würde JAN,FEB,MAR beispielsweise Januar, Februar und März enthalten.

- Das Sonderzeichen - (Bindestrich) gibt Bereiche an. Im Feld "Tag" steht 1-15 für die Tage 1 bis 15 des angegebenen Monats.
- Das Sonderzeichen * (Sternchen) steht für alle Werte im Feld. Im Feld für die Stundenangaben steht * für alle Stunden. Sie können * nicht sowohl in den ay-of-week Feldern D ay-of-month als auch in D in demselben Ausdruck verwenden. Wenn Sie es in einem der Felder eingeben, müssen Sie im anderen Feld ein ? verwenden.
- Das Zeichen / (Schrägstrich) steht für schrittweise Steigerungen. Im Feld Minuten können Sie 1/10 eingeben, um jede zehnte Minute anzugeben, beginnend mit der ersten Minute der Stunde (z. B. die elfte, einundzwanzigste und einunddreißigste Minute usw.).
- Das Zeichen ? (Fragezeichen) steht für einen Wert. Wenn Sie 7 in das ay-of-month Feld D eingeben und es Ihnen egal ist, welcher Wochentag der siebte ist, können Sie eingeben? im ay-of-week D-Feld.
- Der Platzhalter L in den ay-of-week Feldern D ay-of-month oder D gibt den letzten Tag des Monats oder der Woche an.
- Der **`W`** Platzhalter im ay-of-month D-Feld gibt einen Wochentag an. **`3W`**Gibt im ay-of-month Feld D den Wochentag an, der dem dritten Tag des Monats am nächsten liegt.
- Der Platzhalter # im ay-of-week Feld D gibt eine bestimmte Instanz des angegebenen Wochentags innerhalb eines Monats an. 3#2 ist beispielsweise der zweite Dienstag im Monat. Die 3 bezieht sich auf Dienstag, da dies der dritte Tag jeder Woche ist, und die 2 bezieht sich auf den zweiten Tag dieses Typs innerhalb des Monats.

Einschränkungen

- Sie können die ay-of-week Felder D ay-of-month und D nicht im selben Cron-Ausdruck angeben. Wenn Sie in einem der Felder einen Wert oder ein `*` (Sternchen) angeben, müssen Sie ein ? (Fragezeichen) im anderen.
- cron-Ausdrücke werden mit einer Ausführungsrate ab einer Minute unterstützt, kürzere Intervalle sind nicht möglich.
- Sie können einen Canary nicht so einstellen, dass er länger als ein Jahr wartet, bevor er ausgeführt wird. Sie können also nur `*` im `Year`-Feld angeben.

Beispiele

Wenn Sie einen Canary erstellen, können Sie auf die folgenden Beispiel-Cron-Zeichenfolgen verweisen. Die folgenden Beispiele sind die korrekte Syntax für die Verwendung der AWS SDKs

AWS CLI oder, um einen Canary zu erstellen oder zu aktualisieren. Wenn Sie die CloudWatch Konsole verwenden, lassen Sie das Finale * in jedem Beispiel weg.

Expression	Bedeutung
`0 10 * * ? *`	Ausführung jeden Tag um 10:00 Uhr (UTC)
`15 12 * * ? *`	Ausführung jeden Tag um 12:15 Uhr (UTC)
`0 18 ? * MON-FRI *`	Ausführung jeden Montag bis Freitag um 18:00 Uhr (UTC)
`0 8 1 * ? *`	Ausführung um 8:00 Uhr (UTC) am ersten Tag jedes Monats
`0/10 * ? * MON-SAT *`	Ausführung alle 10 Minuten von Montag bis Samstag jeder Woche
`0/5 8-17 ? * MON-FRI *`	Ausführung alle 5 Minuten von Montag bis Freitag zwischen 08:00 Uhr und 17:55 Uhr (UTC)

Gruppen

Sie können Gruppen erstellen, um Canarys miteinander zu verknüpfen, einschließlich regionsübergreifender Canarys. Die Verwendung von Gruppen kann Ihnen bei der Verwaltung und Automatisierung Ihrer Canarys helfen, und Sie können auch aggregierte Laufergebnisse und Statistiken für alle Canarys in einer Gruppe anzeigen.

Gruppen sind globale Ressourcen. Wenn Sie eine Gruppe erstellen, wird sie in allen AWS Regionen repliziert, die Gruppen unterstützen, und Sie können ihr Kanarienvögel aus jeder dieser Regionen hinzufügen und sie in jeder dieser Regionen anzeigen. Obwohl das Format des Gruppen-ARNs den Namen der Region widerspiegelt, in der sie erstellt wurde, ist eine Gruppe nicht auf eine Region beschränkt. Das bedeutet, dass Sie Canarys aus mehreren Regionen in dieselbe Gruppe einfügen und diese Gruppe dann verwenden können, um all diese Canarys in einer einzigen Ansicht anzuzeigen und zu verwalten.

Gruppen werden in allen Regionen außer den Regionen unterstützt, die standardmäßig deaktiviert sind. Weitere Informationen zu diesen Regionen finden Sie unter Aktivieren einer Region.

Jede Gruppe kann bis zu 10 Canarys enthalten. Sie können bis zu 20 Gruppen in Ihrem Konto haben. Jeder Canary kann Mitglied von bis zu 10 Gruppen sein.

So erstellen Sie eine Gruppe

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Synthetics Canaries aus.
3. Wählen Sie Create Group.
4. Geben Sie unter Group Name (Gruppenname) einen Namen für die Gruppe ein.
5. Wählen Sie Canarys aus, die dieser Gruppe zugeordnet werden sollen. Um einen Canary auszuwählen, geben Sie seinen vollständigen Namen in Exact canary name (Eindeutiger Canary-Name) und wählen Sie Search (Suche). Aktivieren Sie dann das Kontrollkästchen neben dem Canary-Namen. Wenn es in verschiedenen Regionen mehrere Canarys mit demselben Namen gibt, achten Sie darauf, die gewünschten Canarys auszuwählen.

   Sie können diesen Schritt wiederholen, um bis zu 10 Canarys der Gruppe zuzuordnen.
6. (Optional) Fügen Sie unter Tags ein oder mehrere Schlüssel/Wert-Paare als Tags für diese Gruppe hinzu. Mithilfe von Tags können Sie Ihre AWS Ressourcen identifizieren und organisieren und Ihre Kosten nachverfolgen. AWS Weitere Informationen finden Sie unter Verschlagworten Sie Ihre Amazon-Ressourcen CloudWatch .
7. Wählen Sie Create Group.

Testen Sie einen Kanarienvogel vor Ort

In diesem Abschnitt wird erklärt, wie Sie CloudWatch Synthetics Canaries direkt im Microsoft Visual Studio Code-Editor oder Code-Editor ändern, testen und debuggen können. JetBrains IDE Die lokale Debugging-Umgebung verwendet einen SAM-Container (Serverless Application Model), um eine Lambda-Funktion zu simulieren, die das Verhalten eines Synthetics-Canaries emuliert.

> **Note**
>
> Es ist nicht praktikabel, Canaries, die auf visueller Überwachung basieren, lokal zu debuggen. Bei der visuellen Überwachung werden während eines ersten Durchlaufs Basis-Screenshots aufgenommen und diese Screenshots dann mit den Screenshots aus nachfolgenden Durchläufen verglichen. In einer lokalen Entwicklungsumgebung werden Läufe nicht gespeichert oder nachverfolgt, und jede Iteration ist ein unabhängiger, eigenständiger Lauf. Das Fehlen eines Canary-Run-Verlaufs macht es unpraktisch, Canaries zu debuggen, die auf visuelle Überwachung angewiesen sind.

Voraussetzungen

1. Wählen oder erstellen Sie einen Amazon S3 S3-Bucket zum Speichern von Artefakten aus lokalen Canary-Testläufen, wie HAR-Dateien und Screenshots. Dazu muss Ihnen IAM bereitgestellt werden. Wenn Sie die Einrichtung von Amazon S3 S3-Buckets überspringen, können Sie Ihren Canary trotzdem lokal testen. Sie erhalten jedoch eine Fehlermeldung über den fehlenden Bucket und Sie haben keinen Zugriff auf Canary-Artefakte.

   Wenn Sie einen Amazon S3 S3-Bucket verwenden, empfehlen wir Ihnen, den Bucket-Lebenszyklus so einzustellen, dass Objekte nach einigen Tagen gelöscht werden, um Kosten zu sparen. Weitere Informationen finden Sie unter Verwalten Ihres Speicher-Lebenszyklus.
2. Richten Sie ein AWS Standardprofil für Ihr AWS Konto ein. Weitere Informationen finden Sie unter Konfiguration und Einstellungen für Anmeldeinformationsdateien.
3. Stellen Sie die AWS Standardregion der Debug-Umgebung auf Ihre bevorzugte Region ein, z. B. `us-west-2`
4. Installieren Sie die AWS SAM CLI. Weitere Informationen finden Sie unter Installation der AWS SAM CLI.
5. Installieren Sie Visual Studio Code Editor oderJetBrains IDE. Weitere Informationen finden Sie unter Visual Studio Codeoder JetBrains IDE
6. Installieren SieDocker, um mit der AWS SAM CLI zu arbeiten. Stellen Sie sicher, dass Sie den Docker-Daemon starten. Weitere Informationen finden Sie unter Installation Docker zur Verwendung mit der AWS SAM CLI.

   Alternativ können Sie auch andere Container-Management-Software installierenRancher, z. B., sofern sie die Docker Runtime verwendet.
7. Installieren Sie eine AWS Toolkit-Erweiterung für Ihren bevorzugten Editor. Weitere Informationen finden Sie unter Installation von AWS Toolkit for Visual Studio Code oder Installation von. AWS Toolkit for JetBrains

Themen

- Richten Sie die Test- und Debugging-Umgebung ein
- Verwenden von Visual Studio Code IDE
- Verwenden von JetBrains IDE
- Führen Sie einen Canary lokal mit der SAM-CLI aus
- Integrieren Sie Ihre lokale Testumgebung in ein vorhandenes Canary-Paket

- Ändern Sie die CloudWatch Synthetics-Laufzeit
- Häufige Fehler

Richten Sie die Test- und Debugging-Umgebung ein

Klonen Sie zunächst das Github-Repository, das AWS Ihnen zur Verfügung steht, indem Sie den folgenden Befehl eingeben. Das Repository enthält Codebeispiele sowohl für Node.js Canaries als auch für Python Canaries.

```
git clone https://github.com/aws-samples/synthetics-canary-local-debugging-sample.git
```

Führen Sie dann, abhängig von der Sprache Ihrer Kanaren, einen der folgenden Schritte aus.

Für Node.js Kanarienvögel

1. Gehen Sie zum Canary-Quellverzeichnis von Node.js, indem Sie den folgenden Befehl eingeben.

   ```
   cd synthetics-canary-local-debugging-sample/nodejs-canary/src
   ```

2. Geben Sie den folgenden Befehl ein, um Canary-Abhängigkeiten zu installieren.

   ```
   npm install
   ```

Für Python-Kanarienvögel

1. Gehen Sie zum Python-Canary-Quellverzeichnis, indem Sie den folgenden Befehl eingeben.

   ```
   cd synthetics-canary-local-debugging-sample/python-canary/src
   ```

2. Geben Sie den folgenden Befehl ein, um Canary-Abhängigkeiten zu installieren.

   ```
   pip3 install -r requirements.txt -t .
   ```

Verwenden von Visual Studio Code IDE

Die Visual Studio Startkonfigurationsdatei befindet sich unter. `.vscode/launch.json` Sie enthält Konfigurationen, mit denen die Vorlagendatei durch isual Studio V-Code erkannt werden kann.

Es definiert eine Lambda-Payload mit den erforderlichen Parametern, um den Canary erfolgreich aufzurufen. Hier ist die Startkonfiguration für einen Node.js Canary:

```
{
            ...
            ...
            "lambda": {
                "payload": {
                    "json": {
                        // Canary name. Provide any name you like.
                        "canaryName": "LocalSyntheticsCanary",
                        // Canary artifact location
                        "artifactS3Location": {
                            "s3Bucket": "cw-syn-results-123456789012-us-west-2",
                            "s3Key": "local-run-artifacts",
                        },
                        // Your canary handler name
                        "customerCanaryHandlerName": "heartbeat-canary.handler"
                    }
                },
                // Environment variables to pass to the canary code
                "environmentVariables": {}
            }
        }
    ]
}
```

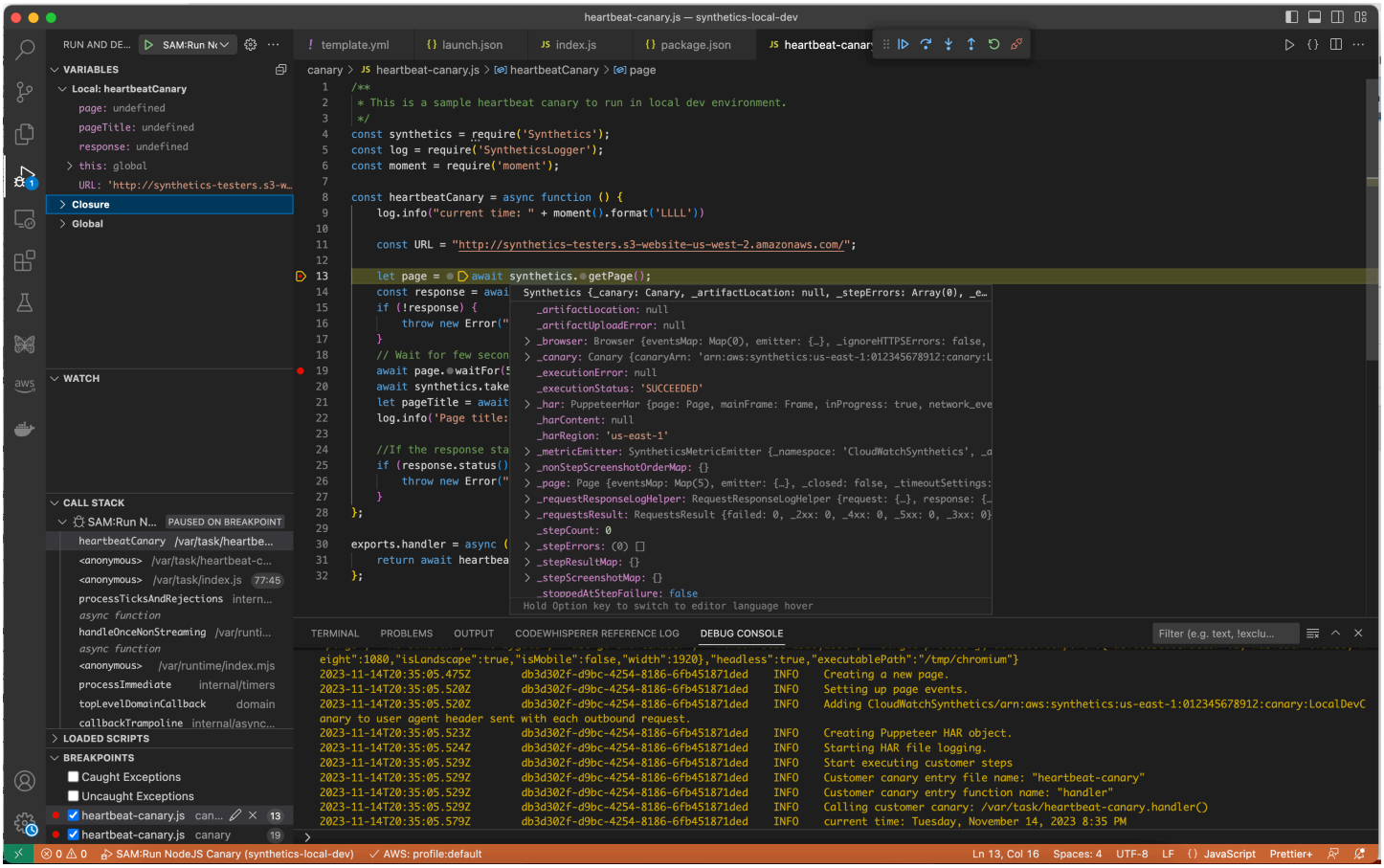
Sie können optional auch die folgenden Felder in der Payload-JSON angeben:

- `s3EncryptionMode`Gültige Werte: | `SSE_S3` `SSE_KMS`
- `s3KmsKeyArn`Gültiger Wert: *`KMS Key ARN`*
- `activeTracing`Gültige Werte: `true` | `false`
- `canaryRunId`Gültiger Wert: *`UUID`* Dieser Parameter ist erforderlich, wenn die aktive Ablaufverfolgung aktiviert ist.

Um den Canary in zu debuggenVisual Studio, fügen Sie dem Canary-Code Breakpoints hinzu, an denen Sie die Ausführung unterbrechen möchten. Um einen Breakpoint hinzuzufügen, wählen Sie den Rand des Editors und wechseln Sie im Editor in den Ausführungs- und Debug-Modus. Starte den Canary, indem du auf die Play-Schaltfläche klickst. Wenn der Canary läuft, werden die Protokolle in der Debug-Konsole protokolliert, sodass Sie in Echtzeit Einblicke in das Verhalten des

Canary erhalten. Wenn Sie Breakpoints hinzugefügt haben, wird die Canary-Ausführung an jedem Breakpoint angehalten, sodass Sie den Code Schritt für Schritt durchgehen und Variablenwerte, Instanzmethoden, Objektattribute und die Funktionsaufrufliste überprüfen können.

Für das lokale Ausführen und Debuggen von Canaries fallen keine Kosten an, mit Ausnahme der im Amazon S3 S3-Bucket gespeicherten Artefakte und der bei jedem lokalen Lauf generierten CloudWatch Metriken.



Verwenden von JetBrains IDE

Nachdem Sie die AWS Toolkit for JetBrains Erweiterung installiert haben, stellen Sie sicher, dass das Plug-In und der JavaScript Debugger Node.js für die Ausführung aktiviert sind, wenn Sie einen Node.js Canary debuggen. Gehen Sie dann folgendermaßen vor.

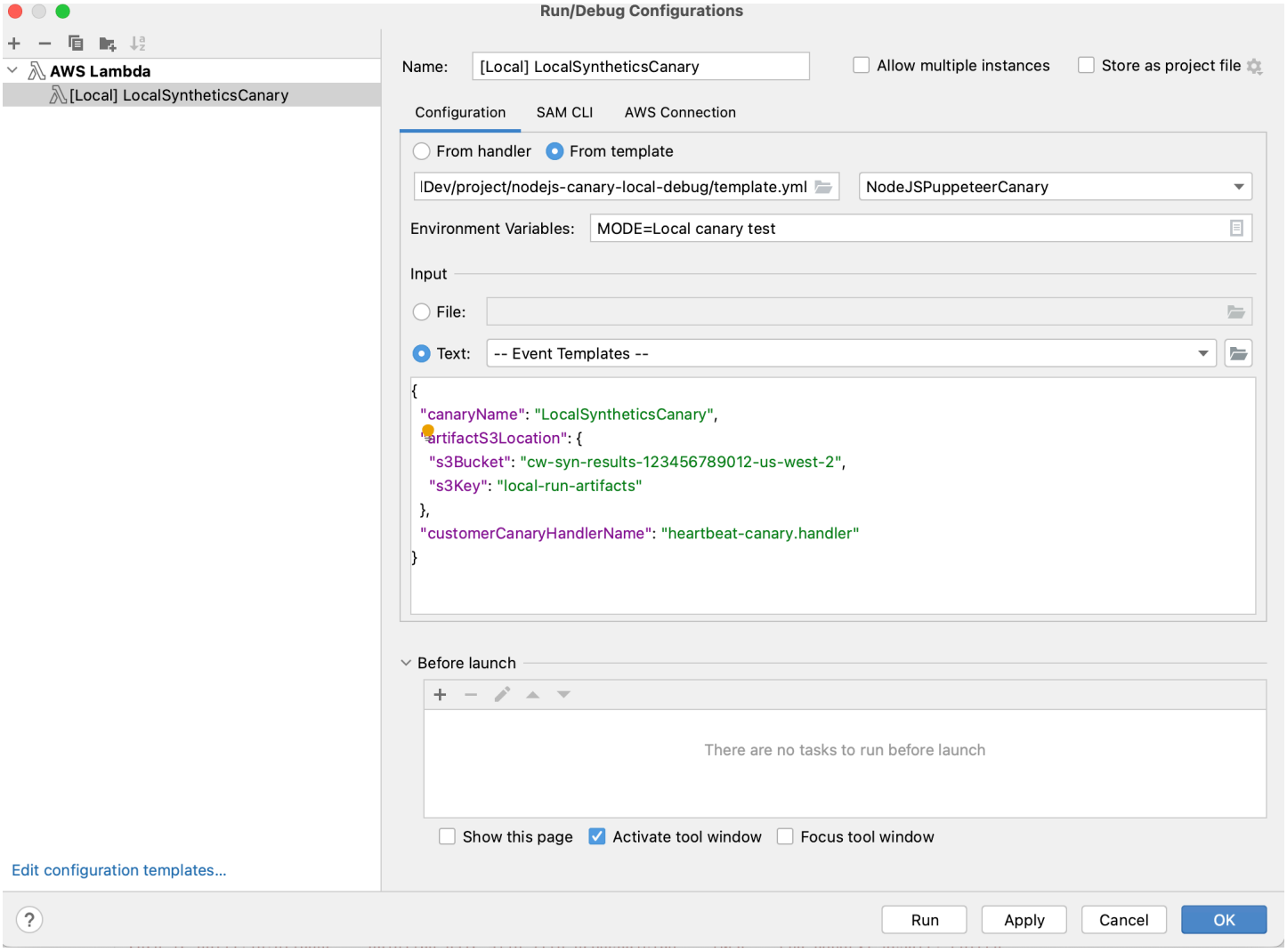
Debuggen Sie einen Canary mit JetBrains IDE

1. Wählen Sie im linken Navigationsbereich von JetBrains IDE Lambda und dann die lokale Konfigurationsvorlage aus.
2. Geben Sie einen Namen für die Ausführungskonfiguration ein, z. B. **`LocalSyntheticsCanary`**

3. Wählen Sie Aus Vorlage, wählen Sie den Dateibrowser im Vorlagenfeld und wählen Sie dann die Datei template.yml aus dem Projekt aus, entweder aus dem Verzeichnis nodejs oder dem Python-Verzeichnis.
4. Geben Sie im Eingabebereich die Payload für den Canary ein, wie im folgenden Bildschirm gezeigt.

```
{
 "canaryName": "LocalSyntheticsCanary",
 "artifactS3Location": {
     "s3Bucket": "cw-syn-results-123456789012-us-west-2",
     "s3Key": "local-run-artifacts"
 },
 "customerCanaryHandlerName": "heartbeat-canary.handler"
}
```

Sie können auch andere Umgebungsvariablen in der Payload-JSON festlegen, wie unter aufgeführt. Verwenden von Visual Studio Code IDE



Führen Sie einen Canary lokal mit der SAM-CLI aus

Gehen Sie wie folgt vor, um Ihren Canary lokal mit der Serverless Application Model (SAM) CLI auszuführen. Geben Sie unbedingt Ihren eigenen Amazon S3 S3-Bucket-Namen für `s3Bucket` in an `event.json`

So verwenden Sie die SAM-CLI, um einen Node.js Canary auszuführen

1. Gehen Sie zum Quellverzeichnis, indem Sie den folgenden Befehl eingeben.

```
cd synthetics-canary-local-debugging-sample/nodejs-canary
```

2. Geben Sie die folgenden Befehle ein.

```
sam build
```

```
sam local invoke -e ../event.json
```

So verwenden Sie die SAM-CLI, um einen Python-Canary auszuführen

1. Gehen Sie zum Quellverzeichnis, indem Sie den folgenden Befehl eingeben.

```
cd synthetics-canary-local-debugging-sample/python-canary
```

2. Geben Sie die folgenden Befehle ein.

```
sam build
sam local invoke -e ../event.json
```

Integrieren Sie Ihre lokale Testumgebung in ein vorhandenes Canary-Paket

Sie können das lokale Canary-Debugging in Ihr bestehendes Canary-Paket integrieren, indem Sie drei Dateien kopieren:

- Kopiere die `template.yml` Datei in das Stammverzeichnis deines Canary-Paketes. Achten Sie darauf, den Pfad so `CodeUri` zu ändern, dass er auf das Verzeichnis verweist, in dem Ihr Canary-Code existiert.
- Wenn du mit einem Node.js Canary arbeitest, kopiere die `cw-synthetics.js` Datei in dein Canary-Quellverzeichnis. Wenn Sie mit einem Python-Canary arbeiten, kopieren Sie `cw-synthetics.py` den in Ihr Canary-Quellverzeichnis.
- Kopieren Sie die Startkonfigurationsdatei. `vscode/launch.json`in das Paketstammverzeichnis. Stellen Sie sicher, dass Sie es im `.vscode` Verzeichnis ablegen; erstellen Sie es, falls es noch nicht existiert.

Ändern Sie die CloudWatch Synthetics-Laufzeit

Im Rahmen Ihres Debuggings möchten Sie vielleicht versuchen, einen Canary mit einer anderen CloudWatch Synthetics-Laufzeit anstelle der neuesten Runtime auszuführen. Suchen Sie dazu in einer der folgenden Tabellen nach der Runtime, die Sie verwenden möchten. Achten Sie darauf, die Laufzeit für die richtige Region auszuwählen. Fügen Sie dann den ARN für diese Laufzeit an der entsprechenden Stelle in Ihrer `template.yml` Datei ein und führen Sie dann den Canary aus.

Node.js-Laufzeiten

ARNs für syn-nodejs-puppeteer -7.0

In der folgenden Tabelle sind die ARNs aufgeführt, die für die Version `syn-nodejs-puppeteer-7.0` der CloudWatch Synthetics-Laufzeit in jeder AWS Region verwendet werden sollen, in der sie verfügbar ist.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:44`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:46`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:44`
USA West (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:47`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:44`
Asien-Pazifik (Hongkong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:45`
Asien-Pazifik (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:20`
Asien-Pazifik (Jakarta)	`arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:26`
Asien-Pazifik (Melbourne)	`arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:18`
Asien-Pazifik (Mumbai)	`arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:44`

Region	ARN
Asia Pacific (Osaka)	arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:30
Asia Pacific (Seoul)	arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:46
Asien-Pazifik (Singapur)	arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:49
Asien-Pazifik (Sydney)	arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:44
Asien-Pazifik (Tokio)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:44
Canada (Central)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:44
Kanada West (Calgary)	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:76
China (Peking)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:45
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:46
Europe (Frankfurt)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:44
Europa (Irland)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:46
Europa (London)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:44
Europa (Milan)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:45

Region	ARN
Europa (Paris)	`arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:44`
Europa (Spain)	`arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:20`
Europa (Stockholm)	`arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:44`
Europa (Zürich)	`arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:19`
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:17`
Naher Osten (Bahrain)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:44`
Naher Osten (VAE)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:19`
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:45`
AWS GovCloud (US-Ost)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:41`
AWS GovCloud (US-West)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:42`

RANs für -6,2 syn-nodejs-puppeteer

In der folgenden Tabelle sind die ARNs aufgeführt, die für die Version `syn-nodejs-puppeteer-6.2` der CloudWatch Synthetics-Laufzeit in jeder AWS Region verwendet werden sollen, in der sie verfügbar ist.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:41`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:43`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:41`
USA West (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:44`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:41`
Asien-Pazifik (Hongkong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:42`
Asien-Pazifik (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:17`
Asien-Pazifik (Jakarta)	`arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:23`
Asien-Pazifik (Melbourne)	`arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:15`
Asien-Pazifik (Mumbai)	`arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:41`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:27`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:42`

Region	ARN
Asien-Pazifik (Singapur)	arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:46
Asien-Pazifik (Sydney)	arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:41
Asien-Pazifik (Tokio)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:41
Canada (Central)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:41
Kanada West (Calgary)	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:73
China (Peking)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:42
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:43
Europe (Frankfurt)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:41
Europa (Irland)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:43
Europa (London)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:41
Europa (Milan)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:42
Europa (Paris)	arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:41
Europa (Spain)	arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:17

Region	ARN
Europa (Stockholm)	`arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:41`
Europa (Zürich)	`arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:16`
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:14`
Naher Osten (Bahrain)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:41`
Naher Osten (VAE)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:16`
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:42`
AWS GovCloud (US-Ost)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:39`
AWS GovCloud (US-West)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:39`

ARNs für -5.2 syn-nodejs-puppeteer

In der folgenden Tabelle sind die ARNs aufgeführt, die für die Version `syn-nodejs-puppeteer-5.2` der CloudWatch Synthetics-Laufzeit in jeder AWS Region verwendet werden sollen, in der sie verfügbar ist.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:42`

Region	ARN
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:44`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:42`
USA West (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:45`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:42`
Asien-Pazifik (Hongkong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:43`
Asien-Pazifik (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:18`
Asien-Pazifik (Jakarta)	`arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:24`
Asien-Pazifik (Melbourne)	`arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:16`
Asien-Pazifik (Mumbai)	`arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:42`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:28`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:44`
Asien-Pazifik (Singapur)	`arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:47`
Asien-Pazifik (Sydney)	`arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:42`

Region	ARN
Asien-Pazifik (Tokio)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:42
Canada (Central)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:42
Kanada West (Calgary)	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:74
China (Peking)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:43
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:44
Europe (Frankfurt)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:42
Europa (Irland)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:44
Europa (London)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:42
Europa (Milan)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:43
Europa (Paris)	arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:42
Europa (Spain)	arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:18
Europa (Stockholm)	arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:42
Europa (Zürich)	arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:17

Region	ARN
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:15`
Naher Osten (Bahrain)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:42`
Naher Osten (VAE)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:17`
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:43`
AWS GovCloud (US-Ost)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:40`
AWS GovCloud (US-West)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:40`

Python-Laufzeiten

ARNs für -3,0 syn-python-selenium

In der folgenden Tabelle sind die ARNs aufgeführt, die für die Version `syn-python-selenium-3.0` der CloudWatch Synthetics-Laufzeit in jeder AWS Region verwendet werden sollen, in der sie verfügbar ist.

Region	ARN
USA Ost (Nord-Virginia)	`aarn:aws:lambda:us-east-1:378653112637:layer:Synthetics_Selenium:32`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics_Selenium:34`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics_Selenium:32`

Region	ARN
USA West (Oregon)	arn:aws:lambda:us-west-2:760325925879:layer:Synthetics_Selenium:34
Africa (Cape Town)	arn:aws:lambda:af-south-1:461844272066:layer:Synthetics_Selenium:32
Asien-Pazifik (Hongkong)	arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics_Selenium:32
Asien-Pazifik (Hyderabad)	arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics_Selenium:20
Asien-Pazifik (Jakarta)	arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics_Selenium:26
Asien-Pazifik (Melbourne)	arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics_Selenium:18
Asien-Pazifik (Mumbai)	arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics_Selenium:32
Asia Pacific (Osaka)	arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics_Selenium:30
Asia Pacific (Seoul)	arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics_Selenium:34
Asien-Pazifik (Singapur)	arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics_Selenium:37
Asien-Pazifik (Sydney)	arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics_Selenium:32
Asien-Pazifik (Tokio)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics_Selenium:32
Canada (Central)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics_Selenium:32

Region	ARN
Kanada West (Calgary)	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics_Selenium:76
China (Peking)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics_Selenium:32
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics_Selenium:32
Europe (Frankfurt)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics_Selenium:32
Europa (Irland)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics_Selenium:34
Europa (London)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics_Selenium:32
Europa (Milan)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics_Selenium:33
Europa (Paris)	arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics_Selenium:32
Europa (Spain)	arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics_Selenium:20
Europa (Stockholm)	arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics_Selenium:32
Europa (Zürich)	arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics_Selenium:19
Israel (Tel Aviv)	arn:aws:lambda:il-central-1:313249807427:layer:Synthetics_Selenium:17
Naher Osten (Bahrain)	arn:aws:lambda:me-south-1:823195537320:layer:Synthetics_Selenium:32

Region	ARN
Naher Osten (VAE)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics_Selenium:19`
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics_Selenium:33`
AWS GovCloud (US-Ost)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics_Selenium:30`
AWS GovCloud (US-West)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics_Selenium:31`

ARNs für -2.1 syn-python-selenium

In der folgenden Tabelle sind die ARNs aufgeführt, die für die Version `syn-python-selenium-2.1` der CloudWatch Synthetics-Laufzeit in jeder AWS Region verwendet werden sollen, in der sie verfügbar ist.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:29`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:31`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:29`
USA West (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:31`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:29`

Region	ARN
Asien-Pazifik (Hongkong)	arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:29
Asien-Pazifik (Hyderabad)	arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:17
Asien-Pazifik (Jakarta)	arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:23
Asien-Pazifik (Melbourne)	arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:15
Asien-Pazifik (Mumbai)	arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:29
Asia Pacific (Osaka)	arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:27
Asia Pacific (Seoul)	arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:30
Asien-Pazifik (Singapur)	arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:34
Asien-Pazifik (Sydney)	arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:29
Asien-Pazifik (Tokio)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:29
Canada (Central)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:29
Kanada West (Calgary)	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:73
China (Peking)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:29

Region	ARN
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:29
Europe (Frankfurt)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:29
Europa (Irland)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:31
Europa (London)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:29
Europa (Milan)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:30
Europa (Paris)	arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:29
Europa (Spain)	arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:17
Europa (Stockholm)	arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:29
Europa (Zürich)	arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:16
Israel (Tel Aviv)	arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:14
Naher Osten (Bahrain)	arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:29
Naher Osten (VAE)	arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:16
Südamerika (São Paulo)	arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:30

Region	ARN
AWS GovCloud (US-Ost)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:29`
AWS GovCloud (US-West)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:29`

Häufige Fehler

Fehler: Für die lokale Ausführung von AWS SAM-Projekten ist Docker erforderlich. Hast du es installiert und läuft?

Stellen Sie sicher, dass Sie Docker auf Ihrem Computer starten.

Lokaler SAM-Aufruf ist fehlgeschlagen: Beim Aufrufen des GetLayerVersion Vorgangs ist ein Fehler aufgetreten (ExpiredTokenException): Das in der Anfrage enthaltene Sicherheitstoken ist abgelaufen

Stellen Sie sicher, dass das AWS Standardprofil eingerichtet ist.

Häufigere Fehler

Weitere Informationen zu häufigen Fehlern im Zusammenhang mit dem SAM finden Sie unter AWS SAM CLI Troubleshooting.

Problembehandlung bei fehlgeschlagenem Canary

Wenn Ihr Canary fehlschlägt, gehen Sie wie folgt vor:

Allgemeine Problembehebung

- Verwenden Sie die Canarydetailseite, um weitere Informationen zu erhalten. Wählen Sie in der CloudWatch Konsole im Navigationsbereich Canaries und dann den Namen des Canaries aus, um die Canary-Detailseite zu öffnen. Prüfen Sie auf der Registerkarte „Verfügbarkeit“ anhand der SuccessPercentMetrik, ob das Problem ständig oder nur sporadisch auftritt.

  Wählen Sie auf der Registerkarte Verfügbarkeit einen fehlgeschlagenen Datenpunkt aus, um Screenshots, Protokolle und Schrittberichte (sofern verfügbar) für diese fehlgeschlagene Ausführung anzuzeigen.

Wenn ein Schrittbericht verfügbar ist, weil Schritte Teil des Skripts sind, überprüfen Sie, welcher Schritt fehlgeschlagen ist, und sehen Sie sich die zugehörigen Screenshots an, um das Problem anzuzeigen, das Ihren Kunden auftritt.

Sie können auch die HAR-Dateien überprüfen, um festzustellen, ob eine oder mehrere Anforderungen fehlschlagen. Sie können tiefer graben, indem Sie Protokolle verwenden, um fehlgeschlagene Anforderungen und Fehler anzuzeigen. Schließlich können Sie diese Artefakte mit den Artefakten eines erfolgreichen Canary-Laufs vergleichen, um das Problem zu ermitteln.

Standardmäßig erfasst CloudWatch Synthetics Screenshots für jeden Schritt in einem UI-Kanarium. Ihr Skript ist jedoch möglicherweise so konfiguriert, dass Screenshots deaktiviert werden. Während des Debuggings möchten Sie möglicherweise Screenshots erneut aktivieren. In ähnlicher Weise möchten Sie für API-Canarys möglicherweise HTTP-Anforderungs- und Antwort-Header und Texte während des Debuggings sehen. Informationen zum Einschließen dieser Daten in den Bericht finden Sie unter executeHttpStep(stepName, RequestOptions, [Rückruf], [StepConfig]).

- Wenn Sie eine kürzlich bereitgestellte Anwendung hatten, rollen Sie sie zurück und debuggen Sie sie später.
- Stellen Sie manuell eine Verbindung zu Ihrem Endpunkt her, um zu sehen, ob Sie das gleiche Problem reproduzieren können.

Themen

- Fehlerbehebung für ein Canary in einer VPC

Canary schlägt nach dem Update der Lambda-Umgebung fehl

CloudWatch Synthetics Canaries sind als Lambda-Funktionen in Ihrem Konto implementiert. Diese Lambda-Funktionen unterliegen regelmäßigen Lambda-Runtime-Updates, die Sicherheitsupdates, Bugfixes und andere Verbesserungen enthalten. Lambda ist bestrebt, Runtime-Updates bereitzustellen, die mit bestehenden Funktionen abwärtskompatibel sind. Wie beim Software-Patching gibt es jedoch seltene Fälle, in denen sich eine Laufzeitaktualisierung negativ auf eine vorhandene Funktion auswirken kann. Wenn Sie glauben, dass Ihr Canary von einem Lambda-Runtime-Update betroffen ist, können Sie den manuellen Modus für Lambda-Laufzeitmanagement (in unterstützten Regionen) verwenden, um die Lambda-Laufzeitversion vorübergehend zurückzusetzen. Dadurch bleibt Ihre Canary-Funktion funktionsfähig und Störungen werden minimiert, sodass Sie Zeit haben, die Inkompatibilität zu beheben, bevor Sie zur neuesten Runtime-Version zurückkehren.

Wenn dein Canary nach einem Lambda-Runtime-Update ausfällt, ist die beste Lösung ein Upgrade auf eine der neuesten Synthetics-Laufzeiten. Weitere Informationen zu den neuesten Laufzeiten finden Sie unter. Synthetics Laufzeitversionen

Als alternative Lösung können Sie in Regionen, in denen Lambda-Laufzeitmanagement-Steuerelemente verfügbar sind, einen Canary auf eine ältere von Lambda verwaltete Laufzeit zurücksetzen, indem Sie den manuellen Modus für die Laufzeitverwaltungssteuerung verwenden. Sie können den manuellen Modus entweder mithilfe der AWS CLI oder mithilfe der Lambda-Konsole einrichten, indem Sie die folgenden Schritte in den folgenden Abschnitten ausführen.

> **Warning**
>
> Wenn Sie die Laufzeiteinstellungen in den manuellen Modus ändern, erhält Ihre Lambda-Funktion keine automatischen Sicherheitsupdates, bis sie wieder in den automatischen Modus zurückversetzt wird. Während dieses Zeitraums kann Ihre Lambda-Funktion anfällig für Sicherheitslücken sein.

Voraussetzungen

- Installieren Sie jq
- Installieren Sie die neueste Version von. AWS CLI Weitere Informationen finden Sie in den AWS CLI Installations- und Aktualisierungsanweisungen.

Schritt 1: Rufen Sie die Lambda-Funktion ARN ab

Führen Sie den folgenden Befehl aus, um das `EngineArn` Feld aus der Antwort abzurufen. Dies `EngineArn` ist der ARN der Lambda-Funktion, die dem Canary zugeordnet ist. Sie werden diesen ARN in den folgenden Schritten verwenden.

```
aws synthetics get-canary --name my-canary | jq '.Canary.EngineArn'
```

Beispielausgabe von`EngingArn`:

```
"arn:aws:lambda:us-west-2:123456789012:function:cwsyn-my-canary-dc5015c2-db17-4cb5-
afb1-EXAMPLE991:8"
```

Schritt 2: Besorgen Sie sich den ARN der letzten guten Lambda-Laufzeitversion

Um zu verstehen, ob Ihr Canary von einem Lambda-Runtime-Update betroffen war, überprüfen Sie, ob Datum und Uhrzeit der ARN-Änderungen der Lambda-Runtime-Version in Ihren Logs dem Datum und der Uhrzeit entsprechen, zu der Sie Auswirkungen auf Ihren Canary festgestellt haben. Wenn sie nicht übereinstimmen, ist es wahrscheinlich kein Lambda-Runtime-Update, das Ihre Probleme verursacht.

Wenn Ihr Canary von einem Lambda-Runtime-Update betroffen ist, müssen Sie den ARN der funktionierenden Lambda-Runtime-Version identifizieren, die Sie zuvor verwendet haben. Folgen Sie den Anweisungen unter Identifizieren von Laufzeitversionsänderungen, um den ARN der vorherigen Laufzeit zu ermitteln. Notieren Sie den ARN der Runtime-Version und fahren Sie mit Schritt 3 fort, um die Runtime-Management-Konfiguration festzulegen.

Wenn Ihr Canary noch nicht von einem Lambda-Umgebungsupdate betroffen ist, können Sie den ARN der Lambda-Laufzeitversion finden, die Sie derzeit verwenden. Führen Sie den folgenden Befehl aus, um die `RuntimeVersionArn` Lambda-Funktion aus der Antwort abzurufen.

```
aws lambda get-function-configuration \
--function-name "arn:aws:lambda:us-west-2:123456789012:function:cwsyn-my-canary-
dc5015c2-db17-4cb5-afb1-EXAMPLE991:8" | jq '.RuntimeVersionConfig.RuntimeVersionArn'
```

Beispielausgabe von`RuntimeVersionArn`:

```
"arn:aws:lambda:us-
west-2::runtime:EXAMPLE647b82f490a45d7ddd96b557b916a30128d9dcab5f4972911ec0f"
```

Schritt 3: Aktualisierung der Lambda Runtime Management-Konfiguration

Sie können entweder die AWS CLI oder die Lambda-Konsole verwenden, um die Runtime-Management-Konfiguration zu aktualisieren.

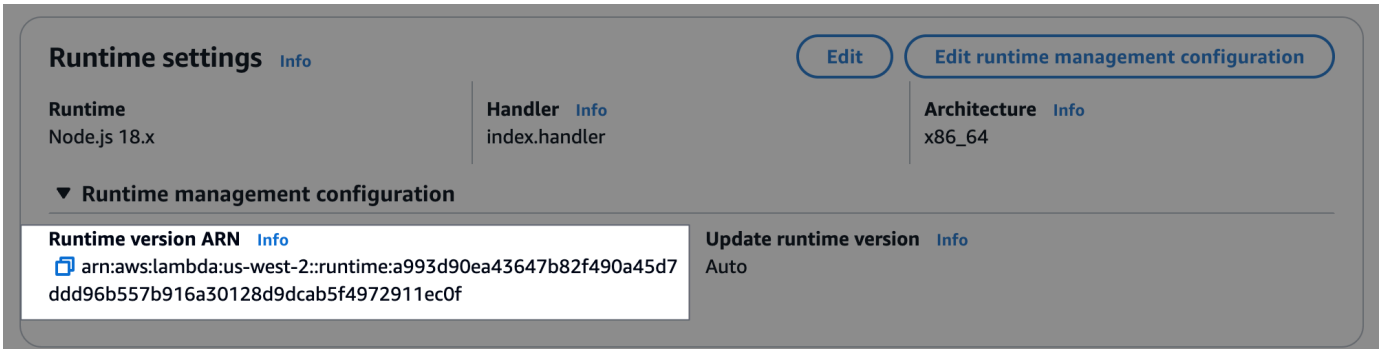
Um den manuellen Modus für die Konfiguration des Lambda-Laufzeitmanagements einzustellen, verwenden Sie den AWS CLI

Geben Sie den folgenden Befehl ein, um das Laufzeitmanagement der Lambda-Funktion in den manuellen Modus zu ändern. Achten Sie darauf, den *Funktionsnamen* und den *Qualifier* durch den ARN der Lambda-Funktion bzw. die Versionsnummer der Lambda-Funktion zu ersetzen, indem Sie die Werte verwenden, die Sie in Schritt 1 gefunden haben. Ersetzen Sie auch die *runtime-version-arn*durch die Version ARN, die Sie in Schritt 2 gefunden haben.

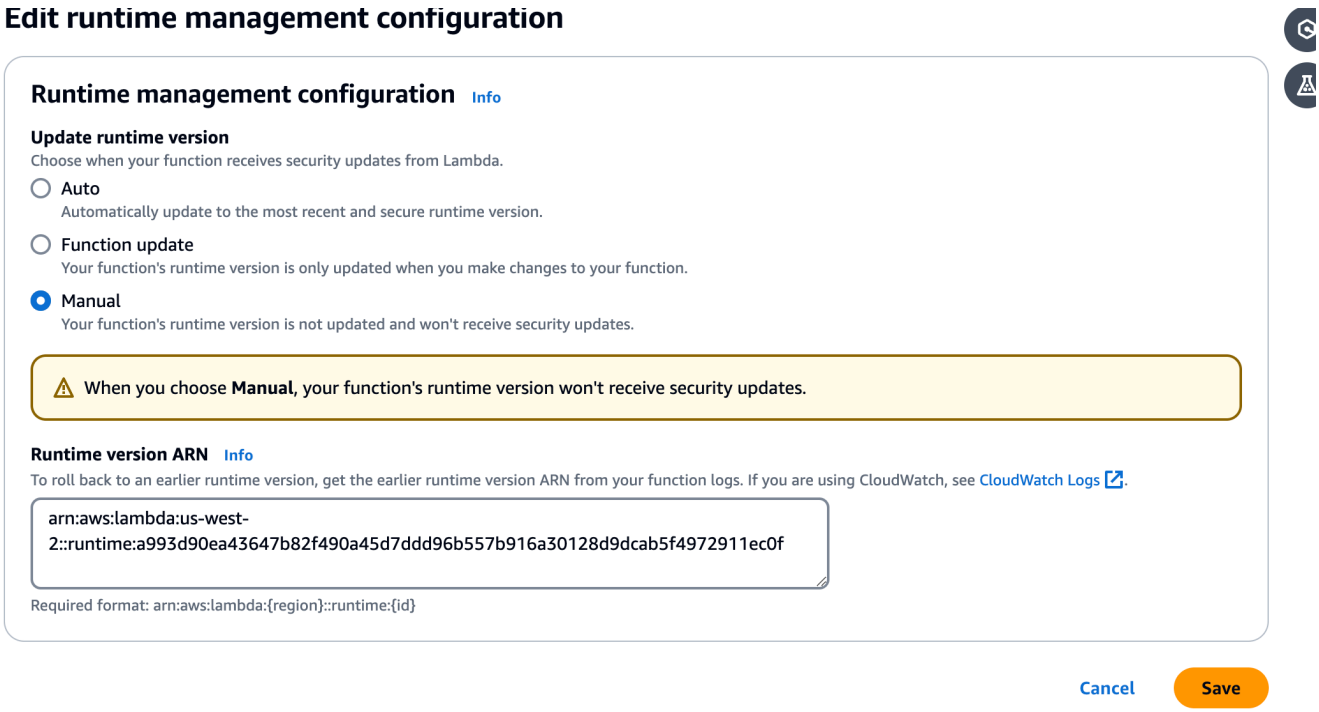
```
aws lambda put-runtime-management-config \
    --function-name "arn:aws:lambda:us-west-2:123456789012:function:cwsyn-my-canary-dc5015c2-db17-4cb5-afb1-EXAMPLE991" \
    --qualifier 8 \
    --update-runtime-on "Manual" \
    --runtime-version-arn "arn:aws:lambda:us-west-2::runtime:a993d90ea43647b82f490a45d7ddd96b557b916a30128d9dcab5f4972911ec0f"
```

Um einen Canary mithilfe der Lambda-Konsole in den manuellen Modus zu versetzen

1. Öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.
2. Wählen Sie den Tab Versionen, wählen Sie den Link mit der Versionsnummer, der Ihrem ARN entspricht, und wählen Sie den Tab Code.
3. Scrollen Sie nach unten zu Runtime-Einstellungen, erweitern Sie Runtime-Management-Konfiguration und kopieren Sie den ARN der Runtime-Version.



4. Wählen Sie Runtime Management-Konfiguration bearbeiten, wählen Sie Manuell und fügen Sie den zuvor kopierten Runtime-Versions-ARN in das Feld Runtime-Version ein. Wählen Sie dann Speichern.



Mein Canary ist blockiert von AWS WAF

Um zu AWS WAF verhindern, dass dein Canary blockiert wird, richte eine Bedingung für den AWS WAF Zeichenkettenabgleich ein, die die Zeichenfolge erlaubt`CloudWatchSynthetics`. Weitere Informationen findest du in der AWS WAF Dokumentation unter Mit Bedingungen für den Abgleich von Zeichenketten arbeiten.

Warten auf das Erscheinen eines Elements

Wenn Sie nach der Analyse Ihrer Protokolle und Screenshots sehen, dass Ihr Skript darauf wartet, dass ein Element auf dem Bildschirm angezeigt wird und eine Zeitüberschreitung auftritt, überprüfen Sie den entsprechenden Screenshot, um zu sehen, ob das Element auf der Seite angezeigt wird. Überprüfen Sie Ihren `xpath`, um sicherzustellen, dass er richtig ist.

Bei Problemen mit Puppetteer finden Sie auf der Seite von Puppeteer oder in den Internetforen GitHub .

Knoten ist entweder nicht sichtbar oder kein HTMLElement für page.click()

Wenn ein Knoten nicht sichtbar ist oder kein `HTMLElement` für `page.click()` ist, überprüfen Sie zuerst den `xpath`, den Sie zum Klicken auf das Element verwenden. Wenn sich Ihr Element am unteren Bildschirmrand befindet, passen Sie außerdem Ihr Darstellungsfenster an. CloudWatch Synthetics verwendet standardmäßig einen Viewport von 1920 * 1080. Sie können beim Starten des Browsers oder mithilfe der Puppeteer-Funktion `page.setViewport` ein anderes Ansichtsfenster festlegen.

Artifacts können nicht in S3 hochgeladen werden, Ausnahme: S3-Bucket-Speicherort kann nicht abgerufen werden: Zugriff verweigert

Wenn Ihr Canary aufgrund eines Amazon S3 S3-Fehlers ausfällt, konnte CloudWatch Synthetics aufgrund von Berechtigungsproblemen keine Screenshots, Protokolle oder Berichte hochladen, die für den Canary erstellt wurden. Überprüfen Sie, ob Folgendes der Fall ist:

- Prüfen Sie, ob die IAM-Rolle des Canarys die `s3:ListAllMyBuckets`-Berechtigung, die `s3:GetBucketLocation`-Berechtigung für den richtigen Amazon-S3-Bucket und die `s3:PutObject`-Berechtigung für den Bucket besitzt, in dem der Canary seine Artefakte speichert. Wenn der Canary eine visuelle Überwachung durchführt, benötigt die Rolle auch die `s3:GetObject`-Berechtigung für den Bucket. Dieselben Berechtigungen sind auch in der Endpunkt-Richtlinie von Amazon VPC S3 Gateway erforderlich, wenn der Canary in einer VPC mit einem VPC-Endpunkt bereitgestellt wird.
- Wenn der Canary anstelle des standardmäßigen verwalteten Schlüssels (Standard) einen vom AWS KMS Kunden AWS verwalteten Schlüssel für die Verschlüsselung verwendet, ist die IAM-Rolle des Canary möglicherweise nicht berechtigt, mit diesem Schlüssel zu verschlüsseln oder zu entschlüsseln. Weitere Informationen finden Sie unter Verschlüsseln von Canary-Artefakten.
- Ihre Bucket-Richtlinie lässt möglicherweise den Verschlüsselungsmechanismus nicht zu, den der Canary verwendet. Wenn Ihre Bucket-Richtlinie beispielsweise vorschreibt, einen bestimmten Verschlüsselungsmechanismus oder KMS-Schlüssel zu verwenden, müssen Sie denselben Verschlüsselungsmodus für Ihren Canary auswählen.

Führt der Canary eine visuelle Überwachung durch, finden Sie unter Aktualisieren des Artefaktspeicherortes und der Verschlüsselung bei Verwendung der visuellen Überwachung weitere Informationen dazu.

Fehler: Protokollfehler (Runtime). callFunctionOn): Ziel geschlossen.

Dieser Fehler wird angezeigt, wenn nach dem Schließen der Seite oder des Browsers einige Netzwerkanforderungen vorhanden sind. Möglicherweise haben Sie vergessen, auf einen asynchronen Vorgang zu warten. Nach der Ausführung Ihres Skripts schließt CloudWatch Synthetics den Browser. Die Ausführung eines asynchronen Vorgangs nach dem Schließen des Browsers kann dazu führen, dass `target closed error`.

Canary ist fehlgeschlagen. Fehler: Kein Datenpunkt – Canary zeigt Timeout-Fehler

Dies bedeutet, dass Ihr Canarylauf das Timeout überschritten hat. Die Canary-Ausführung wurde gestoppt, bevor CloudWatch Synthetics CloudWatch Erfolgsmetriken in Prozent veröffentlichen oder Artefakte wie HAR-Dateien, Logs und Screenshots aktualisieren konnte. Wenn Ihr Timeout zu niedrig ist, können Sie es erhöhen.

Standardmäßig ist ein Canary-Timeout-Wert gleich seiner Häufigkeit. Sie können den Timeout-Wert manuell so einstellen, dass er kleiner oder gleich der Canary-Frequenz ist. Wenn Ihre Canaryfrequenz niedrig ist, müssen Sie die Frequenz erhöhen, um das Timeout zu erhöhen. Sie können sowohl die Häufigkeit als auch den Timeout-Wert unter Zeitplan anpassen, wenn Sie einen Canary mithilfe der CloudWatch Synthetics-Konsole erstellen oder aktualisieren.

Stellen Sie sicher, dass Ihr canary-Timeout-Wert nicht kürzer als 15 Sekunden ist, um Lambda-Kaltstarts und die Zeit zu ermöglichen, die zum Hochfahren der canary-Instrumentierung benötigt wird.

Canary-Artefakte können in der CloudWatch Synthetics-Konsole nicht angezeigt werden, wenn dieser Fehler auftritt. Du kannst CloudWatch Logs verwenden, um die Logs von Canary einzusehen.

Um CloudWatch Logs zu verwenden, um die Logs eines Kanarienvogels einzusehen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im linken Navigationsbereich Log groups (Protokollgruppen) aus.
3. Suchen Sie die Protokollgruppe, indem Sie den Canary-Namen in das Filterfeld eingeben. Protokollgruppen für Canarys haben den Namen /aws/lambda/cwsyn-***CanaryName***-RandomID.

Versuch, auf einen internen Endpunkt zuzugreifen

Wenn du möchtest, dass dein Canary auf einen Endpunkt in deinem internen Netzwerk zugreift, empfehlen wir dir, CloudWatch Synthetics für die Verwendung von VPC einzurichten. Weitere Informationen finden Sie unter Ausführen eines Canarys in einer VPC.

Probleme beim Upgrade und Downgrade der Canary-Laufzeitversion

Wenn Sie den Canary kürzlich von der Laufzeitversion `syn-1.0` auf eine neuere Version aktualisiert haben, liegt möglicherweise ein Problem mit der ursprungsübergreifenden Anforderungsfreigabe (CORS) vor. Weitere Informationen finden Sie unter Problem mit Cross-Origin Request Sharing (CORS).

Wenn Sie den Canary kürzlich auf eine ältere Runtime-Version heruntergestuft haben, stellen Sie sicher, dass die CloudWatch Synthetics-Funktionen, die Sie verwenden, in der älteren Runtime-Version verfügbar sind, auf die Sie das Downgrade durchgeführt haben. Die Funktion `executeHttpStep` ist beispielsweise ab der Laufzeitversion `syn-nodejs-2.2` verfügbar. Informationen zur Verfügbarkeit von Funktionen finden Sie unter Ein Canary-Skript schreiben.

> **Note**
>
> Wenn Sie ein Upgrade oder Downgrade der Runtime-Version für einen Canary planen, empfehlen wir Ihnen, zuerst den Canary zu klonen und die Runtime-Version auf dem geklonten Canary zu aktualisieren. Nachdem Sie überprüft haben, dass der Klon mit der neuen Laufzeitversion funktioniert, können Sie die Laufzeitversion Ihres ursprünglichen Canary aktualisieren und den Klon löschen.

Problem mit Cross-Origin Request Sharing (CORS)

Wenn in einem UI-Canary einige Netzwerkanforderungen mit `403` oder `net::ERR_FAILED` fehlschlagen, prüfen Sie, ob für den Canary die aktive Ablaufverfolgung aktiviert ist, und verwendet auch die Puppeteer-Funktion `page.setExtraHTTPHeaders`, um Header hinzuzufügen. Wenn dies der Fall ist, können die fehlgeschlagenen Netzwerkanforderungen durch Cross-Origin Request Sharing (CORS) Einschränkungen verursacht werden. Sie können bestätigen, ob dies der Fall ist, indem Sie die aktive Ablaufverfolgung deaktivieren oder die zusätzlichen HTTP-Header entfernen.

Warum passiert das?

Wenn aktive Ablaufverfolgung verwendet wird, wird allen ausgehenden Anforderungen ein zusätzlicher Header hinzugefügt, um den Aufruf zu verfolgen. Wenn Sie die Anforderungsheader ändern, indem Sie einen Trace-Header hinzufügen oder zusätzliche Header mithilfe von Puppeteer hinzufügen, `page.setExtraHTTPHeaders` führt dies zu einer CORS-Prüfung auf XML- (XHR) - Anfragen. HttpRequest

Wenn Sie das aktive Tracing nicht deaktivieren oder die zusätzlichen Header entfernen möchten, können Sie Ihre Webanwendung aktualisieren, um den ursprungsübergreifenden Zugriff zu ermöglichen, oder Sie können die Websicherheit deaktivieren, indem Sie das Flag `disable-web-security` beim Starten des Chrome-Browsers in Ihrem Skript verwenden.

Mit der Synthetics-Startfunktion können Sie die von CloudWatch Synthetics verwendeten Startparameter überschreiben und zusätzliche `disable-web-security` CloudWatch Flag-Parameter übergeben. Weitere Informationen finden Sie unter Für Node.js-Canary-Skripte verfügbare Bibliotheksfunktionen.

> **Note**
>
> Sie können die von CloudWatch Synthetics verwendeten Startparameter überschreiben, wenn Sie die Laufzeitversion `syn-nodejs-2.1` oder höher verwenden.

Probleme mit den Bedingungen des Canary-Rennens

Um die beste Erfahrung mit CloudWatch Synthetics zu erzielen, stellen Sie sicher, dass der für die Kanaren geschriebene Code idempotent ist. Andernfalls kann es in seltenen Fällen bei Canary-Runs zu Rennbedingungen kommen, wenn der Canary bei verschiedenen Durchläufen mit derselben Ressource interagiert.

Fehlerbehebung für ein Canary in einer VPC

Wenn nach dem Erstellen oder Aktualisieren eines Canarys auf einem VPC Probleme auftreten, kann einer der folgenden Abschnitte Ihnen helfen, das Problem zu beheben.

Neues Canary im Fehlerzustand oder Canary kann nicht aktualisiert werden

Wenn Sie ein Canary zur Ausführung in einer VPC erstellen und dieses sofort in einen Fehlerzustand geht oder Sie ein Canary nicht so aktualisieren können, dass es in einer VPC ausgeführt wird, verfügt die Rolle des Canarys möglicherweise nicht über die korrekten

Berechtigungen. Um auf einer VPC ausgeführt werden zu können, muss ein Canary über die Berechtigungen `ec2:CreateNetworkInterface`, `ec2:DescribeNetworkInterfaces` und `ec2:DeleteNetworkInterface` verfügen. Diese Berechtigungen sind alle in der von `AWSLambdaVPCAccessExecutionRole` verwalteten Richtlinie enthalten. Weitere Informationen finden Sie unter Ausführungsrolle und Benutzerberechtigungen.

Wenn dieses Problem beim Erstellen eines Canarys aufgetreten ist, müssen Sie das Canary löschen und ein neues erstellen. Wenn Sie die CloudWatch Konsole verwenden, um den neuen Canary zu erstellen, wählen Sie unter Zugriffsberechtigungen die Option Neue Rolle erstellen aus. Es wird eine neue Rolle erstellt, die alle für die Ausführung des Canary erforderlichen Berechtigungen enthält.

Wenn dieses Problem auftritt, wenn Sie ein Canary aktualisieren, können Sie das Canary erneut aktualisieren und eine neue Rolle bereitstellen, die über die erforderlichen Berechtigungen verfügt.

Fehler „No test result returned“ (Kein Testergebnis zurückgegeben)

Wenn ein Canary den Fehler „No test result returned (Kein Testergebnis zurückgegeben)“ anzeigt, kann eines der folgenden Probleme die Ursache sein:

- Wenn Ihre VPC keinen Internetzugang hat, müssen Sie VPC-Endpunkte verwenden, um dem Canary Zugriff auf Amazon S3 zu CloudWatch gewähren. Sie müssen die Optionen für die DNS resolution (DNS-Auflösung) und den DNS hostname (DNS-Hostnamen) in der VPC aktivieren, damit diese Endpunktadressen korrekt aufgelöst werden können. Weitere Informationen finden Sie unter Verwenden von DNS mit Ihrer VPC CloudWatch und Verwenden von CloudWatch Synthetics mit Schnittstellen-VPC-Endpunkten.
- Canaries müssen in privaten Subnetzen innerhalb einer VPC ausgeführt werden. Um dies zu überprüfen, öffnen Sie die Seite Subnets (Subnetze) in der VPC-Konsole. Überprüfen Sie die Subnetze, die Sie bei der Konfiguration des Canarys ausgewählt haben. Wenn diese einen Pfad zu einem Internet-Gateway (igw-) haben, sind sie keine privaten Subnetze.

Informationen zur Behebung dieser Probleme finden Sie in den Protokollen für das Canary.

So zeigen Sie die Protokollereignisse von einem Canary an:

1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/cloudwatch/ CloudWatch .
2. Wählen Sie im Navigationsbereich Protokollgruppen aus.
3. Wählen Sie den Namen der Protokollgruppe des Canarys. Der Name der Protokollgruppe beginnt mit `/aws/lambda/cwsyn-`*`canary-name`*.

Beispielcode für Canary-Skripte

Dieser Abschnitt enthält Codebeispiele, die einige mögliche Funktionen für Canary-Skripte von CloudWatch Synthetics veranschaulichen.

Beispiele für Node.js und Puppeteer

Festlegen von Cookies

Websites verlassen sich auf Cookies, um benutzerdefinierte Funktionen bereitzustellen oder Benutzer zu verfolgen. Durch das Setzen von Cookies in CloudWatch Synthetics-Skripten können Sie dieses benutzerdefinierte Verhalten nachahmen und validieren.

Beispielsweise kann eine Website für einen erneut besuchenden Benutzer anstelle eines Registrierungslinks einen Anmeldungs-Link anzeigen.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const pageLoadBlueprint = async function () {

    let url = "http://smile.amazon.com/";

    let page = await synthetics.getPage();

    // Set cookies.  I found that name, value, and either url or domain are required
 fields.
    const cookies = [{
      'name': 'cookie1',
      'value': 'val1',
      'url': url
    },{
      'name': 'cookie2',
      'value': 'val2',
      'url': url
    },{
      'name': 'cookie3',
      'value': 'val3',
      'url': url
    }];

    await page.setCookie(...cookies);
```

```
    // Navigate to the url
    await synthetics.executeStep('pageLoaded_home', async function (timeoutInMillis =
 30000) {

        var response = await page.goto(url, {waitUntil: ['load', 'networkidle0'],
 timeout: timeoutInMillis});

        // Log cookies for this page and this url
        const cookiesSet = await page.cookies(url);
        log.info("Cookies for url: " + url + " are set to: " +
 JSON.stringify(cookiesSet));
    });

};

exports.handler = async () => {
    return await pageLoadBlueprint();
};
```

Emulation des Geräts

Sie können Skripte schreiben, die verschiedene Geräte emulieren, sodass Sie annähernd das Aussehen und das Verhalten einer Seite auf diesen Geräten ermitteln können.

Das folgende Beispiel emuliert ein iPhone 6-Gerät. Weitere Informationen zur Emulation finden Sie unter page.emulate(options) in der Puppeteer-Dokumentation.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');
const puppeteer = require('puppeteer-core');

const pageLoadBlueprint = async function () {

    const iPhone = puppeteer.devices['iPhone 6'];

    // INSERT URL here
    const URL = "https://amazon.com";

    let page = await synthetics.getPage();
    await page.emulate(iPhone);

    //You can customize the wait condition here. For instance,
    //using 'networkidle2' may be less restrictive.
```

```
    const response = await page.goto(URL, {waitUntil: 'domcontentloaded', timeout:
 30000});
    if (!response) {
        throw "Failed to load page!";
    }

    await page.waitFor(15000);

    await synthetics.takeScreenshot('loaded', 'loaded');

    //If the response status code is not a 2xx success code
    if (response.status() < 200 || response.status() > 299) {
        throw "Failed to load page!";
    }
};

exports.handler = async () => {
    return await pageLoadBlueprint();
};
```

Mehrstufiger API-Canary

Dieser Beispielcode veranschaulicht einen API-Canary mit zwei HTTP-Schritten: Testen der gleichen API für positive und negative Testfälle. Die Schrittkonfiguration wird übergeben, um das Reporting von Anfrage/Antwort-Headern zu ermöglichen. Darüber hinaus blendet es den Authorization Header und X-AMZ-Security-Token aus, da sie Benutzeranmeldeinformationen enthalten.

Wenn dieses Skript als Canary verwendet wird, können Sie Details zu jedem Schritt und den zugehörigen HTTP-Anforderungen anzeigen, z. B. Schritt-Pass/Fail-Wert, Dauer und Performance-Metriken wie DNS-Nachschlagezeit und erste Byte-Zeit. Sie können die Anzahl der 2xx, 4xx und 5xx für Ihren Canarylauf anzeigen.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const apiCanaryBlueprint = async function () {

    // Handle validation for positive scenario
    const validatePositiveCase = async function(res) {
        return new Promise((resolve, reject) => {
            if (res.statusCode < 200 || res.statusCode > 299) {
                throw res.statusCode + ' ' + res.statusMessage;
```

```
            }

            let responseBody = '';
            res.on('data', (d) => {
                responseBody += d;
            });

            res.on('end', () => {
                // Add validation on 'responseBody' here if required. For ex, your
 status code is 200 but data might be empty
                resolve();
            });
        });
    };

    // Handle validation for negative scenario
    const validateNegativeCase = async function(res) {
        return new Promise((resolve, reject) => {
            if (res.statusCode < 400) {
                throw res.statusCode + ' ' + res.statusMessage;
            }

            resolve();
        });
    };

    let requestOptionsStep1 = {
        'hostname': 'myproductsEndpoint.com',
        'method': 'GET',
        'path': '/test/product/validProductName',
        'port': 443,
        'protocol': 'https:'
    };

    let headers = {};
    headers['User-Agent'] = [synthetics.getCanaryUserAgentString(), headers['User-
Agent']].join(' ');

    requestOptionsStep1['headers'] = headers;

    // By default headers, post data and response body are not included in the report
 for security reasons.
    // Change the configuration at global level or add as step configuration for
 individual steps
```

```
    let stepConfig = {
        includeRequestHeaders: true,
        includeResponseHeaders: true,
        restrictedHeaders: ['X-Amz-Security-Token', 'Authorization'], // Restricted
 header values do not appear in report generated.
        includeRequestBody: true,
        includeResponseBody: true
    };


    await synthetics.executeHttpStep('Verify GET products API with valid name',
 requestOptionsStep1, validatePositiveCase, stepConfig);

    let requestOptionsStep2 = {
        'hostname': 'myproductsEndpoint.com',
        'method': 'GET',
        'path': '/test/canary/InvalidName(',
        'port': 443,
        'protocol': 'https:'
    };

    headers = {};
    headers['User-Agent'] = [synthetics.getCanaryUserAgentString(), headers['User-
Agent']].join(' ');

    requestOptionsStep2['headers'] = headers;

    // By default headers, post data and response body are not included in the report
 for security reasons.
    // Change the configuration at global level or add as step configuration for
 individual steps
    stepConfig = {
        includeRequestHeaders: true,
        includeResponseHeaders: true,
        restrictedHeaders: ['X-Amz-Security-Token', 'Authorization'], // Restricted
 header values do not appear in report generated.
        includeRequestBody: true,
        includeResponseBody: true
    };

    await synthetics.executeHttpStep('Verify GET products API with invalid name',
 requestOptionsStep2, validateNegativeCase, stepConfig);

};
```

```
exports.handler = async () => {
    return await apiCanaryBlueprint();
};
```

Beispiele für Python und Selenium

Der folgende Selenium-Beispielcode ist ein Canary, der mit einer benutzerdefinierten Fehlermeldung fehlschlägt, wenn ein Zielelement nicht geladen wird.

```
from aws_synthetics.selenium import synthetics_webdriver as webdriver
from aws_synthetics.common import synthetics_logger as logger
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By

def custom_selenium_script():
    # create a browser instance
    browser = webdriver.Chrome()
    browser.get('https://www.example.com/')
    logger.info('navigated to home page')
    # set cookie
    browser.add_cookie({'name': 'foo', 'value': 'bar'})
    browser.get('https://www.example.com/')
    # save screenshot
    browser.save_screenshot('signed.png')
    # expected status of an element
    button_condition = EC.element_to_be_clickable((By.CSS_SELECTOR, '.submit-button'))
    # add custom error message on failure
    WebDriverWait(browser, 5).until(button_condition, message='Submit button failed to
 load').click()
    logger.info('Submit button loaded successfully')
    # browser will be quit automatically at the end of canary run,
    # quit action is not necessary in the canary script
    browser.quit()

# entry point for the canary
def handler(event, context):
    return custom_selenium_script()
```

Canary- und X-Ray-Ablaufverfolgung

Sie können wählen, ob Sie die aktive AWS X-Ray Ablaufverfolgung auf Kanaren aktivieren möchten, die Runtime oder eine spätere Laufzeit verwenden. `syn-nodejs-2.0` Wenn Tracing aktiviert ist, werden Traces für alle Aufrufe des Canary gesendet, die den Browser, das AWS SDK oder HTTP- oder HTTPS-Module verwenden. Canarys mit aktiviertem Tracing werden auf der X-Ray Trace Map und in Application Signals angezeigt, nachdem Sie es für Ihre Anwendung aktiviert haben.

> **Note**
>
> Das Aktivieren der X-Ray-Verfolgung auf canaries wird in der Region Asien-Pazifik (Jakarta) noch nicht unterstützt.

Wenn ein Canary auf der X-Ray Trace Map angezeigt wird, wird er als neuer Client-Knotentyp angezeigt. Sie können den Mauszeiger auf einen Canary-Knoten bewegen, um Daten über Latenz, Anforderungen und Fehler anzuzeigen. Sie können auch den Canary-Knoten auswählen, um weitere Daten unten auf der Seite zu sehen. In diesem Bereich der Seite kannst du „In Synthetics anzeigen“ wählen, um zur Synthetics-Konsole zu springen, um weitere Informationen über den CloudWatch Canary zu erhalten, oder „Traces anzeigen“ wählen, um mehr Details zu den Traces aus den Läufen dieses Canary zu sehen.

Ein Canary mit aktivierten Ablaufverfolgungen hat auf seiner Detailseite auch eine Registerkarte Tracing mit Details zu Ablaufverfolgungen und Segmenten aus den Läufen des Canarys.

Durch Aktivieren der Ablaufverfolgung wird die Canary-Laufzeit um 2,5 % auf 7 % erhöht.

Ein Canary mit aktivierter Ablaufverfolgung muss eine Rolle mit den folgenden Berechtigungen verwenden. Wenn Sie die Konsole verwenden, um die Rolle beim Erstellen des Canary zu erstellen, erhalten Sie diese Berechtigungen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Sid230934",
            "Effect": "Allow",
            "Action": [
                "xray:PutTraceSegments"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

Von Canarys erzeugte Spuren sind gebührenpflichtig. Weitere Informationen zu den Preisen für X-Ray finden Sie unter AWS X-Ray – Preise.

Ausführen eines Canarys in einer VPC

Sie können Canarys auf Endpunkten in einer VPC und auf öffentlichen internen Endpunkten ausführen. Um ein Canary auf einer VPC auszuführen, müssen sowohl die Optionen DNS Resolution (DNS-Auflösung) als auch DNS hostnames (DNS-Hostnamen) auf der VPC aktiviert sein. Weitere Informationen finden Sie unter Using DNS with Your VPC.

Wenn Sie einen Canary auf einem VPC-Endpunkt ausführen, müssen Sie ihm eine Möglichkeit bieten, seine Metriken CloudWatch und seine Artefakte an Amazon S3 zu senden. Wenn die VPC bereits für den Internetzugang aktiviert ist, brauchen Sie nichts weiter zu tun. Das Canary wird in Ihrer VPC ausgeführt, kann aber auf das Internet zugreifen, um seine Metriken und Artefakte hochzuladen.

Wenn die VPC noch nicht für den Internetzugriff aktiviert ist, haben Sie zwei Möglichkeiten:

- Aktivieren Sie sie für den Internetzugriff. Weitere Informationen finden Sie im folgenden Abschnitt: Mit einer VPC Internetzugriff auf Ihren Canary gewähren.
- Wenn Sie Ihre VPC privat halten möchten, können Sie den Canary so konfigurieren, dass er seine Daten über private VPC-Endpunkte an CloudWatch und Amazon S3 sendet. Falls Sie dies noch nicht getan haben, müssen Sie einen VPC-Endpunkt für CloudWatch (com.amazonaws) erstellen. *region* .monitoring) und einen Gateway-Endpunkt für Amazon S3. Weitere Informationen finden Sie unter Verwendung von CloudWatch und CloudWatch Synthetics mit VPC-Endpunkten mit Schnittstelle und Amazon VPC-Endpunkte für Amazon S3.

Mit einer VPC Internetzugriff auf Ihren Canary gewähren

Folge diesen Schritten, um deinem VPC Canary Internetzugang zu gewähren oder deinem Canary eine statische IP-Adresse zuzuweisen

Internetzugang zu einem Canary auf einer VPC gewähren

1. Erstellen Sie ein NAT-Gateway in einem öffentlichen Subnetz auf der VPC. Detaillierte Anweisungen finden Sie unter Create a NAT gateway (NAT-Gateway erstellen).
2. Fügen Sie der Routentabelle im privaten Subnetz, in dem der Canary gestartet wird, eine neue Route hinzu. Machen Sie folgende Angaben:

   - Geben Sie für Destination (Ziel) **`0.0.0.0/0`** ein.
   - Wählen Sie für Ziel NAT Gateway aus und wählen Sie dann die ID des NAT-Gateways, das Sie erstellt haben.
   - Wählen Sie Save Rules (Routen speichern) aus.

   Weitere Informationen über das Hinzufügen von Routen finden Sie unter Add and remove routes form a route table (Hinzufügen und Entfernen von Routen aus einer Routing-Tabelle).

> **Note**
>
> Stellen Sie sicher, dass der Status der Routen zu Ihrem NAT-Gateway aktiv ist. Wenn das NAT-Gateway gelöscht wird und Sie die Routen nicht aktualisiert haben, haben sie einen Blackhole-Status. Weitere Informationen hierzu finden Sie unter Work with NAT gateways (Arbeiten mit NAT-Gateways).

Verschlüsseln von Canary-Artefakten

CloudWatch Synthetics speichert kanarische Artefakte wie Screenshots, HAR-Dateien und Berichte in Ihrem Amazon S3 S3-Bucket. Standardmäßig werden diese Artefakte im Ruhezustand mit einem AWS verwalteten Schlüssel verschlüsselt. Weitere Informationen finden Sie unter Kundenschlüssel und AWS Schlüssel.

Sie können wählen, ob Sie eine andere Verschlüsselungsoption verwenden möchten. CloudWatch Synthetics unterstützt Folgendes:

- SSE S3 – Serverseitige Verschlüsselung (SSE) mit einem von Amazon S3 verwalteten Schlüssel.
- SSE-KMS – Serverseitige Verschlüsselung (SSE) mit einem vom Kunden verwalteten AWS KMS -Schlüssel.

Wenn Sie die Standardverschlüsselungsoption mit einem AWS verwalteten Schlüssel verwenden möchten, benötigen Sie keine zusätzlichen Berechtigungen.

Um SSE-S3-Verschlüsselung zu verwenden, geben Sie SSE_S3 als Verschlüsselungsmodus an, wenn Sie einen Canary erstellen oder aktualisieren. Um diesen Verschlüsselungsmodus zu verwenden, benötigen Sie keine zusätzlichen Berechtigungen. Weitere Informationen finden Sie unter Protecting data using server-side encryption with Amazon S3-managed encryption keys (SSE-S3) (Schutz von Daten mithilfe serverseitiger Verschlüsselung mit Amazon S3-verwalteten Verschlüsselungsschlüsseln (SSE-S3).

Um einen vom AWS KMS Kunden verwalteten Schlüssel zu verwenden, geben Sie SSE-KMS als Verschlüsselungsmodus an, wenn Sie Ihren Canary erstellen oder aktualisieren, und Sie geben auch den Amazon-Ressourcennamen (ARN) Ihres Schlüssels an. Sie können auch einen kontoübergreifenden KMS-Schlüssel verwenden.

Um einen vom Kunden verwalteten Schlüssel verwenden zu können, benötigen Sie folgende Einstellungen:

- Die IAM-Rolle für den Canary muss die Berechtigung haben, Ihre Artefakte mit Ihrem Schlüssel zu verschlüsseln. Wenn Sie eine visuelle Überwachung verwenden, müssen Sie ihr auch die Berechtigung zum Entschlüsseln von Artefakten erteilen.

```
{
    "Version": "2012-10-17",
    "Statement": {"Effect": "Allow",
        "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
        ],
        "Resource": "Your KMS key ARN"
    }
}
```

- Anstatt Ihrer IAM-Rolle Berechtigungen hinzuzufügen, können Sie Ihre IAM-Rolle zu Ihrer Schlüsselrichtlinie hinzufügen. Wenn Sie dieselbe Rolle für mehrere Canarys verwenden, sollten Sie diesen Ansatz in Betracht ziehen.

```
{
    "Sid": "Enable IAM User Permissions",
    "Effect": "Allow",
    "Principal": {
```

```
            "AWS": "Your synthetics IAM role ARN"
        },
        "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
        ],
        "Resource": "*"
}
```

- Wenn Sie einen kontoübergreifenden KMS-Schlüssel verwenden, lesen Sie den Abschnitt Gestatten, dass Benutzer anderer Konten einen KMS-Schlüssel verwenden.

Anzeigen verschlüsselter Canary-Artefakte bei Verwendung eines vom Kunden verwalteten Schlüssels

Um Canary-Artefakte anzuzeigen, aktualisieren Sie Ihren vom Kunden verwalteten Schlüssel, sodass der Benutzer, der sich AWS KMS die Artefakte ansieht, die Entschlüsselungsberechtigung erhält. Alternativ können Sie dem Benutzer oder der IAM-Rolle, die die Artefakte anzeigt, Entschlüsselungsberechtigungen hinzufügen.

Die AWS KMS Standardrichtlinie aktiviert IAM-Richtlinien im Konto, um den Zugriff auf die KMS-Schlüssel zu ermöglichen. Wenn Sie einen kontoübergreifenden KMS-Schlüssel verwenden, finden Sie weitere Informationen unter Warum erhalten kontoübergreifende Benutzer die Fehlermeldung „Zugriff verweigert“, wenn sie versuchen, auf Amazon S3 S3-Objekte zuzugreifen, die mit einem benutzerdefinierten AWS KMS Schlüssel verschlüsselt wurden? .

Weitere Informationen zum Beheben von Problemen mit Zugriffsverweigerung aufgrund eines KMS-Schlüssels finden Sie unter Troubleshooting key access (Fehlerbehebung beim Zugriff mit einem Schlüssel).

Aktualisieren des Artefaktspeicherortes und der Verschlüsselung bei Verwendung der visuellen Überwachung

Um eine visuelle Überwachung durchzuführen, vergleicht CloudWatch Synthetics Ihre Screenshots mit Basis-Screenshots, die in dem als Baseline ausgewählten Lauf aufgenommen wurden. Wenn Sie Ihren Artefaktspeicherort oder Ihre Verschlüsselungsoption aktualisieren, müssen Sie einen der folgenden Schritte ausführen:

- Stellen Sie sicher, dass Ihre IAM-Rolle sowohl für den vorherigen Amazon-S3-Standort als auch für den neuen Amazon-S3-Standort für Artefakte über ausreichende Berechtigungen

verfügt. Stellen Sie außerdem sicher, dass sie sowohl für die vorherigen als auch für die neuen Verschlüsselungsmethoden und KMS-Schlüssel berechtigt ist.

- Erstellen Sie eine neue Baseline, indem Sie die nächste Canary-Ausführung als neue Baseline auswählen. Wenn Sie diese Option verwenden, müssen Sie nur sicherstellen, dass Ihre IAM-Rolle über ausreichende Berechtigungen für den neuen Artefaktspeicherort und die Verschlüsselungsoption verfügt.

Wir empfehlen die zweite Möglichkeit, die nächste Ausführung als neue Baseline auszuwählen. Dies vermeidet eine Abhängigkeit von einem Artefaktspeicherort oder einer Verschlüsselungsoption, die Sie nicht mehr für den Canary verwenden.

Angenommen, Ihr Canary verwendet Artefaktposition A und KMS-Schlüssel K zum Hochladen von Artefakten. Wenn Sie Ihren Canary auf den Artefaktstandort B und KMS-Schlüssel L aktualisieren, können Sie sicherstellen, dass Ihre IAM-Rolle Berechtigungen sowohl für die Artefaktpositionen (A und B) als auch für beide KMS-Schlüssel (K und L) hat. Alternativ können Sie die nächste Ausführung als neue Baseline auswählen und sicherstellen, dass Ihre Canary-IAM-Rolle Berechtigungen für den Artefaktspeicherort B und KMS-Schlüssel L hat.

Anzeigen von Canary-Statistiken und -Details

Sie können Details zu Ihren Canaries anzeigen und Statistiken über ihre Ausführungen einsehen.

Um alle Details zu Ihren Canary-Ausführungen sehen zu können, müssen Sie bei einem Konto angemeldet sein, das über ausreichende Berechtigungen verfügt. Weitere Informationen finden Sie unter Erforderliche Rollen und Berechtigungen für CloudWatch Kanarienvögel.

So zeigen Sie Canary-Statistiken und -Details an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Synthetics Canaries aus.

   In den Details zu den von Ihnen erstellten Canarys:

   - Status zeigt visuell an, wie viele Ihrer Canarys ihre letzten Ausführungen bestanden haben.
   - Groups (Gruppen) zeigt die von Ihnen erstellten Gruppen an und zeigt an, wie viele von ihnen ausgefallene oder alarmierende Canarys haben.
   - Slowest performers (Langsamste Performer) zeigt die Gruppe und die Region mit den Canarys mit der langsamsten Leistung an. Diese werden berechnet, indem die durchschnittliche

Dauer aller Canarys (über die ausgewählte Zeitspanne) innerhalb einer Gruppe oder Region addiert und durch die Anzahl der Canarys in der Gruppe oder Region dividiert wird. Wenn Sie die Metrik für Langsamste Gruppe auswählen, wird die Tabelle gefiltert, sodass nur die langsamsten Gruppen und ihre Canarys angezeigt werden. Die Tabelle ist nach Durchschnittsdauer sortiert.

- Am Ende der Seite befindet sich eine Tabelle, die alle Canarys anzeigt. Eine Spalte zeigt die Alarme, die für jeden Canary erstellt wurden. Es werden nur Alarme angezeigt, die dem Benennungsstandard für Canaryalarme entsprechen. Dieser Standard ist `Synthetics-Alarm-`*`canaryName`*`-`*`index`* . Canary-Alarme, die Sie im Bereich Synthetics der CloudWatch Konsole erstellen, verwenden automatisch diese Namenskonvention. Wenn Sie Canary-Alarme im Bereich Alarme der CloudWatch Konsole oder mithilfe AWS CloudFormation dieser Namenskonvention erstellen und diese Benennungskonvention nicht verwenden, funktionieren die Alarme, sie werden aber nicht in dieser Liste angezeigt.

3. Um weitere Details zu einem einzelnen Canary anzuzeigen, wählen Sie den Namen des Canarys in der Tabelle Canaries (Canarys) aus.

   In den Details über dieses Canary:

   - Auf der Registerkarte Verfügbarkeit werden Informationen zu den letzten Läufen dieses Canarys angezeigt.

     Unter Canary runs (Canary-Ausführungen) können Sie eine der Zeilen auswählen, um sich Details zu dieser Ausführung anzeigen zu lassen.

     Unter dem Diagramm können Sie Steps (Schritte), Screenshot, Logs (Protokolle), or HAR file (HAR-Datei) auswählen, um diese Arten von Details anzuzeigen. Wenn für den Canary die aktive Ablaufverfolgung aktiviert ist, können Sie auch Ablaufverfolgungen auswählen, um Ablaufverfolgungsinformationen aus den Läufen des Canarys anzuzeigen.

     Die Protokolle für Canary-Läufe werden in S3-Buckets und in CloudWatch Logs gespeichert.

     Screenshots zeigen, wie Ihren Kunden Ihre Webseiten angezeigt werden. Sie können die HAR-Dateien (HTTP-Archivdateien) verwenden, um sich detaillierte Leistungsdaten zu den Webseiten anzeigen zu lassen. Sie können die Liste der Web-Anfragen analysieren und Performance-Probleme wie die Ladezeit für ein Element erfassen. Protokolldateien zeigen die Aufzeichnung der Interaktionen zwischen der Canary-Ausführung und der Webseite und können dazu verwendet werden, detaillierte Angaben zu Fehlern zu identifizieren.

Wenn der Canary die Laufzeit `syn-nodejs-2.0-beta` oder höher verwendet, können Sie die HAR-Dateien nach Statuscode, Anfragegröße oder Dauer sortieren.

Auf der Registerkarte Steps (Schritte) finden Sie eine Liste der Schritte des Canary, den Status jedes Schritts, den Fehlergrund, die URL nach der Ausführung des Schritts, Screenshots und die Dauer der Schrittausführung. Bei API-Canaries mit HTTP-Schritten können Sie Schritte und entsprechende HTTP-Anforderungen anzeigen, wenn Sie die Laufzeit `syn-nodejs-2.2` oder höher verwenden.

Wählen Sie die Registerkarte HTTP-Anfragen, um das Protokoll jeder HTTP-Anfrage anzuzeigen, die vom Canary gestellt wurde. Sie können Anfrage/Antwort-Header, Antworttext, Statuscode, Fehler- und Performance-Timings (Gesamtdauer, TCP-Verbindungszeit, TLS-Handshake-Zeit, erste Byte-Zeit und Inhaltsübertragungszeit) anzeigen. Alle HTTP-Anfragen, die das HTTP/HTTPS-Modul unter der Haube verwenden, werden hier erfasst.

Standardmäßig sind in API-Canarys der Anforderungsheader, der Antwort-Header, der Anforderungskörper und der Antworttext aus Sicherheitsgründen nicht im Bericht enthalten. Wenn Sie sie einbeziehen, werden die Daten nur in Ihrem S3 Bucket gespeichert. Informationen zum Einschließen dieser Daten in den Bericht finden Sie unter executeHttpStep(stepName, RequestOptions, [Rückruf], [StepConfig]).

Die Inhaltstypen des Antworttextkörpers Text, HTML und JSON werden unterstützt. Inhaltstypen wie Text/HTML, Text/Plain, Application/JSON und Application/ -1.0 werden unterstützt. x-amz-json Komprimierte Antworten werden nicht unterstützt.

- Auf der Registerkarte „Überwachung“ werden Grafiken der von diesem Canary veröffentlichten Metriken angezeigt. CloudWatch Weitere Informationen zu diesen Metriken finden Sie unter CloudWatch von Canaries veröffentlichte Metriken.

  Unter den von The Canary veröffentlichten CloudWatch Grafiken befinden sich Diagramme mit Lambda-Metriken, die sich auf den Lambda-Code des Canary beziehen.
- Auf der Registerkarte Konfiguration werden Konfigurations- und Zeitplaninformationen zum Canary angezeigt.
- Das Tab Groups (Gruppen) zeigt die Gruppen an, denen dieser Canary zugeordnet ist, falls vorhanden.
- Auf der Registerkarte Tags werden die Tags angezeigt, die dem Canary zugeordnet sind.

CloudWatch von Canaries veröffentlichte Metriken

Canaries veröffentlichen die folgenden Metriken CloudWatch im Namespace. `CloudWatchSynthetics` Weitere Informationen zum Anzeigen von CloudWatch Metriken finden Sie unter. Anzeigen der verfügbaren Metriken

Metrik	Beschreibung
`SuccessPercent`	Der Prozentsatz der Abläufe dieses Canarys, die erfolgreich sind und keine Fehler finden. Gültige Dimensionen: CanaryName Gültige Statistik: Durchschnitt Einheiten: Prozent
`Duration`	Die Dauer des Canary-Laufs in Millisekunden. Gültige Abmessungen: CanaryName Gültige Statistik: Durchschnitt Einheiten: Millisekunden
`Errors`	Gibt an, wie oft der Canary sein vollständiges Skript nicht ausführen konnte. Gültige Abmessungen: CanaryName Gültige Statistiken: Summe
`2xx`	Die Anzahl der vom Canary ausgeführten Netzwerkanforderungen, die OK-Antworten zurückgegeben haben, mit Antwortcodes zwischen 200 und 299. Diese Metrik wird für UI-Canaries gemeldet, die Laufzeitversion `syn-nodejs-2.0` oder höher verwenden und wird für API-Canar ies gemeldet, die Laufzeitversion `syn-nodejs-2.2` oder höher verwenden.

Metrik	Beschreibung
	Gültige Abmessungen: CanaryName Gültige Statistiken: Summe Einheiten: Anzahl
4xx	Die Anzahl der von Canarys ausgeführten Netzwerkanforderungen, die Fehlerantworten zurückgegeben haben, mit Antwortcodes zwischen 400 und 499. Diese Metrik wird für UI-Canaries gemeldet, die Laufzeitversion `syn-nodejs-2.0` oder höher verwenden und wird für API-Canaries gemeldet, die Laufzeitversion `syn-nodejs-2.2` oder höher verwenden. Gültige Abmessungen: CanaryName Gültige Statistiken: Summe Einheiten: Anzahl
5xx	Die Anzahl der vom Canary ausgeführten Netzwerkanforderungen, die Fehlerantworten zurückgegeben haben, mit Antwortcodes zwischen 500 und 599. Diese Metrik wird für UI-Canaries gemeldet, die Laufzeitversion `syn-nodejs-2.0` oder höher verwenden und wird für API-Canaries gemeldet, die Laufzeitversion `syn-nodejs-2.2` oder höher verwenden. Gültige Abmessungen: CanaryName Gültige Statistiken: Summe Einheiten: Anzahl

Metrik	Beschreibung
`Failed`	Die Anzahl der Canary-Läufe, die nicht ausgeführt werden konnten. Diese Fehler hängen mit dem Canary selbst zusammen. Gültige Abmessungen: CanaryName Gültige Statistiken: Summe Einheiten: Anzahl
`Failed requests`	Die Anzahl der HTTP-Anforderungen, die vom Canary auf der Zielwebsite ausgeführt werden, die ohne Antwort fehlgeschlagen sind. Gültige Abmessungen: CanaryName Gültige Statistiken: Summe Einheiten: Anzahl
`VisualMonitoringSuccessPercent`	Der Prozentsatz der visuellen Vergleiche, die während eines Canarylaufs erfolgreich mit den Grundlinien-Screenshots übereinstimmten. Gültige Abmessungen: CanaryName Gültige Statistik: Durchschnitt Einheiten: Prozent
`VisualMonitoringTotalComparisons`	Die Gesamtzahl der visuellen Vergleiche, die während eines Canarylaufs aufgetreten sind. Gültige Abmessungen: CanaryName Einheiten: Anzahl

> **Note**
>
> Canaries, die entweder die `executeStep()`- oder die `executeHttpStep()`-Methode aus der Synthetics-Bibliothek verwenden, veröffentlichen auch `SuccessPercent`- und `Duration`-Metriken mit den Dimensionen `CanaryName` und `StepName` für jeden Schritt.

Einen Canary bearbeiten oder löschen

Sie können einen vorhandenen Canary bearbeiten oder löschen.

Canary bearbeiten

Wenn Sie einen Canary bearbeiten, wird der Zeitplan entsprechend zurückgesetzt, wenn Sie den Canary bearbeiten. Wenn Sie beispielsweise einen Canary haben, der jede Stunde läuft, und Sie diesen Canary bearbeiten, wird der Canary sofort nach Abschluss der Bearbeitung und dann jede Stunde danach ausgeführt.

So bearbeiten oder aktualisieren Sie einen Canary

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Synthetics Canaries aus.
3. Wählen Sie die Schaltfläche neben dem Canary-Namen und wählen Sie Aktionen, Bearbeiten.
4. (Optional) Wenn dieser Canary eine visuelle Überwachung von Screenshots durchführt und Sie den nächsten Lauf des Canary als Basislinie festlegen möchten, wählen Sie Nächsten Lauf als neue Basislinie festlegen aus.
5. (Optional) Wenn dieser Canary eine visuelle Überwachung von Screenshots durchführt und Sie einen Screenshot aus der visuellen Überwachung entfernen oder Teile des Screenshots festlegen möchten, die bei visuellen Vergleichen ignoriert werden sollen, wählen Sie unter Visuelle Überwachung die Option Basislinie bearbeiten.

   Der Screenshot wird angezeigt, und Sie können einen der folgenden Schritte ausführen:

   - Um die Verwendung des Screenshots für die visuelle Überwachung zu entfernen, wählen Sie Screenshot von der visuellen Testbasislinie entfernen.
   - Um Teile des Screenshots festzulegen, die bei visuellen Vergleichen ignoriert werden sollen, klicken und ziehen, um Bereiche des Bildschirms zu zeichnen, die ignoriert werden sollen.

Wenn Sie dies für alle Bereiche getan haben, die Sie bei Vergleichen ignorieren möchten, wählen Sie Speichern.

6. Nehmen Sie alle anderen gewünschten Änderungen am Canary vor und wählen Sie Speichern.

Canary löschen

Wenn Sie ein Canary löschen, können Sie wählen, ob Sie auch andere Ressourcen löschen möchten, die vom Canary verwendet und erstellt wurden. Wenn Sie ein Canary löschen, sollten Sie auch die folgenden Elemente löschen:

- Lambda-Funktionen und -Ebenen, die von diesem Canary verwendet wurden. Ihr Präfix ist `cwsyn-`*`MyCanaryName`*.
- CloudWatch Alarme, die für diesen Canary erstellt wurden. Diese Alarme haben einen Namen, der mit `Synthetics-Alarm-`*`MyCanaryName`* beginnt. Weitere Informationen zum Löschen von Alarmen finden Sie unter Bearbeiten oder löschen Sie einen CloudWatch Alarm.
- Amazon-S3-Objekte und -Buckets, z. B. den Speicherort der Ergebnisse und Artefakte des Canarys
- IAM-Rollen, die für das Canary erstellt wurden. Diese haben den Namen `role/service-role/CloudWatchSyntheticsRole-`*`MyCanaryName`*.
- Gruppen in CloudWatch Logs protokollieren, die für den Canary erstellt wurden. Diese Protokollgruppen haben folgende Namen: `/aws/lambda/cwsyn-`*`MyCanaryName`*`-`*`randomId`*.

Bevor Sie ein Canary löschen, empfiehlt es sich, sich die Details zum Canary anzeigen zu lassen und diese Informationen zu notieren. Auf diese Weise können Sie die richtigen Ressourcen löschen, nachdem Sie Canary gelöscht haben.

So löschen Sie Canarys

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Synthetics Canaries aus.
3. Wenn der Canary sich derzeit im Status `RUNNING` befindet, müssen Sie ihn anhalten. Canarys können nur in folgenden Status gelöscht werden: `STOPPED`, `READY(NOT_STARTED)` oder `ERROR`.

   Wählen Sie zum Anhalten des Canary die Schaltfläche neben dem Canary-Namen und wählen Sie Actions, (Aktionen) Stop (Stop) aus.

4. Wählen Sie die Schaltfläche neben dem Canary-Namen und wählen Sie Aktionen, Löschen.
5. Wählen Sie aus, ob Sie auch die anderen Ressourcen löschen möchten, die für den Canary erstellt und verwendet werden. Dies umfasst die Lambda-Funktion und Ebenen sowie die IAM-Rolle und die IAM-Richtlinie des Canary.

   Um die IAM-Rolle und die IAM-Richtlinie des Canary zu löschen, müssen Sie über ausreichende Berechtigungen verfügen. Weitere Informationen finden Sie unter AWS verwaltete (vordefinierte) Richtlinien für CloudWatch Synthetics.
6. Geben Sie **`Delete`** in das Feld ein und wählen Sie Löschen.
7. Löschen Sie die anderen Ressourcen, die von dem Canary verwendet und für ihn erstellt wurden, wie oben in diesem Abschnitt aufgeführt.

Laufzeit für mehrere Canary starten, stoppen, löschen oder aktualisieren

Sie können die Laufzeit von bis zu fünf Canary mit einer Aktion stoppen, starten, löschen oder aktualisieren. Wenn Sie die Laufzeit eines Canary aktualisieren, wird diese auf die neueste Laufzeit aktualisiert, die für die Sprache und das Framework verfügbar ist, die der Canary verwendet.

Wenn Sie mehrere Canary auswählen und sich nur einige von ihnen in einem Zustand befinden, der für die von Ihnen gewählte Aktion gültig ist, wird die Aktion nur auf den Canary ausgeführt, für die diese Aktion gültig ist. Wenn Sie beispielsweise einige Canary auswählen, die gerade laufen, und einige, die nicht laufen, und Sie wählen, dass die Canary gestartet werden, werden die Canary, die noch nicht laufen, gestartet, und die Canary, die bereits laufen, sind davon nicht betroffen.

Wenn keiner der ausgewählten Canary für eine Aktion gültig ist, ist diese Aktion nicht im Menü verfügbar.

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Synthetics Canaries aus.
3. Aktivieren Sie die Kontrollkästchen neben den Canary, die Sie anhalten, starten oder löschen möchten.
4. Wählen Sie Actions (Aktionen) und dann entweder Start (Starten), Stop (Anhalten), Delete (Löschen) oder Update Runtime (Laufzeit aktualisieren) aus.

Überwachung kanarischer Ereignisse mit Amazon EventBridge

Die EventBridge Amazon-Veranstaltungsregeln können Sie benachrichtigen, wenn sich der Status der Kanaren ändert oder Läufe abgeschlossen werden. EventBridgeliefert eine Reihe near-real-time von Systemereignissen, die Änderungen an AWS Ressourcen beschreiben. CloudWatch Synthetics sendet diese Ereignisse nach EventBridge bestem Wissen und Gewissen an. Lieferung nach bestem Wissen bedeutet, dass CloudWatch Synthetics versucht, alle Ereignisse an zu senden EventBridge, aber in einigen seltenen Fällen kann es vorkommen, dass eine Veranstaltung nicht zugestellt wird. EventBridge verarbeitet alle empfangenen Ereignisse mindestens einmal. Außerdem empfangen Ihre Ereignis-Listener die Ereignisse möglicherweise nicht in der Reihenfolge, in der die Ereignisse aufgetreten sind.

> **Note**
>
> Amazon EventBridge ist ein Event-Bus-Service, mit dem Sie Ihre Anwendungen mit Daten aus einer Vielzahl von Quellen verbinden können. Weitere Informationen finden Sie unter Was ist Amazon EventBridge? im EventBridge Amazon-Benutzerhandbuch.

CloudWatch Synthetics gibt ein Ereignis aus, wenn ein Canary seinen Status ändert oder einen Lauf abschließt. Sie können eine EventBridge Regel erstellen, die ein Ereignismuster enthält, das allen von CloudWatch Synthetics gesendeten Ereignistypen entspricht, oder die nur bestimmten Ereignistypen entspricht. Wenn ein Canary eine Regel auslöst, EventBridge ruft er die in der Regel definierten Zielaktionen auf. Auf diese Weise können Sie als Reaktion auf eine Canary-Statusänderung oder den Abschluss eines Canary-Laufs Benachrichtigungen senden, Ereignisinformationen erfassen und Korrekturmaßnahmen ergreifen. Sie können beispielsweise Regeln für die folgenden Anwendungsfälle erstellen:

- Untersuchen, wenn ein Canary-Lauf fehlschlägt
- Untersuchen, wann ein Canary in den `ERROR`-Zustand gegangen ist
- Den Lebenszyklus eines Canarys verfolgen
- Überwachen des Erfolgs oder Misserfolgs von Canary-Läufen als Teil eines Workflows

Beispielereignisse von CloudWatch Synthetics

In diesem Abschnitt sind Beispielereignisse von CloudWatch Synthetics aufgeführt. Weitere Informationen zum Ereignisformat finden Sie unter Ereignisse und Ereignismuster in EventBridge.

Canary-Statusänderung

Bei diesem Ereignistyp können die Werte von `current-state` und `previous-state` die folgenden sein:

`CREATING` | `READY` | `STARTING` | `RUNNING` | `UPDATING` | `STOPPING` | `STOPPED` | `ERROR`


```
{
                "version": "0",
                "id": "8a99ca10-1e97-2302-2d64-316c5dedfd61",
                "detail-type": "Synthetics Canary Status Change",
                "source": "aws.synthetics",
                "account": "123456789012",
                "time": "2021-02-09T22:19:43Z",
                "region": "us-east-1",
                "resources": [],
                "detail": {
                                "account-id": "123456789012",
                                "canary-id": "EXAMPLE-dc5a-4f5f-96d1-989b75a94226",
                                "canary-name": "events-bb-1",
                                "current-state": "STOPPED",
                                "previous-state": "UPDATING",
                                "source-location": "NULL",
                                "updated-on": 1612909161.767,
                                "changed-config": {
                                                "executionArn": {
                                                                "previous-value":
 "arn:aws:lambda:us-east-1:123456789012:function:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:1",
                                                                "current-value":
 "arn:aws:lambda:us-east-1:123456789012:function:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:2"
                                                },
                                                "vpcId": {
                                                                "current-value": "NULL"
                                                },
                                                "testCodeLayerVersionArn": {
                                                                "previous-
value": "arn:aws:lambda:us-east-1:123456789012:layer:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:1",
                                                                "current-value":
 "arn:aws:lambda:us-east-1:123456789012:layer:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:2"
                                                }
```

```
                                },
                                "message": "Canary status has changed"
                }
}
```

Erfolgreiche Canary-Ausführung abgeschlossen

```
{
                "version": "0",
                "id": "989EXAMPLE-f4a5-57a7-1a8f-d9cc768a1375",
                "detail-type": "Synthetics Canary TestRun Successful",
                "source": "aws.synthetics",
                "account": "123456789012",
                "time": "2021-02-09T22:24:01Z",
                "region": "us-east-1",
                "resources": [],
                "detail": {
                                "account-id": "123456789012",
                                "canary-id": "989EXAMPLE-dc5a-4f5f-96d1-989b75a94226",
                                "canary-name": "events-bb-1",
                                "canary-run-id": "c6c39152-8f4a-471c-9810-989EXAMPLE",
                                "artifact-location": "cw-syn-results-123456789012-us-
east-1/canary/us-east-1/events-bb-1-ec3-28ddbe266797/2021/02/09/22/23-41-200",
                                "test-run-status": "PASSED",
                                "state-reason": "null",
                                "canary-run-timeline": {
                                                "started": 1612909421,
                                                "completed": 1612909441
                                },
                                "message": "Test run result is generated successfully"
                }
}
```

Fehlerhafte Canary-Ausführung abgeschlossen

```
{
                "version": "0",
                "id": "2644b18f-3e67-5ebf-cdfd-bf9f91392f41",
                "detail-type": "Synthetics Canary TestRun Failure",
                "source": "aws.synthetics",
                "account": "123456789012",
                "time": "2021-02-09T22:24:27Z",
                "region": "us-east-1",
```

```
                "resources": [],
                "detail": {
                                "account-id": "123456789012",
                                "canary-id": "af3e3a05-dc5a-4f5f-96d1-9989EXAMPLE",
                                "canary-name": "events-bb-1",
                                "canary-run-id": "0df3823e-7e33-4da1-8194-
b04e4d4a2bf6",
                                "artifact-location": "cw-syn-results-123456789012-us-
east-1/canary/us-east-1/events-bb-1-ec3-989EXAMPLE/2021/02/09/22/24-21-275",
                                "test-run-status": "FAILED",
                                "state-reason": "\"Error: net::ERR_NAME_NOT_RESOLVED
 \""
                                "canary-run-timeline": {
                                                "started": 1612909461,
                                                "completed": 1612909467
                                },
                                "message": "Test run result is generated successfully"
                }
}
```

Es ist möglich, dass Ereignisse doppelt sind oder nicht in der richtigen Reihenfolge werden. Um die Reihenfolge der Ereignisse zu bestimmen, verwenden Sie die `time`-Eigenschaft.

Voraussetzungen für die Erstellung von EventBridge Regeln

Bevor Sie eine EventBridge Regel für CloudWatch Synthetics erstellen, sollten Sie Folgendes tun:

- Machen Sie sich mit Ereignissen, Regeln und Zielen in EventBridge vertraut.
- Erstellen und konfigurieren Sie die Ziele, die durch Ihre EventBridge Regeln aufgerufen werden. Regeln können viele Arten von Zielen aufrufen, einschließlich:
  - Amazon SNS-Themen
  - AWS Lambda Funktionen
  - Kinesis-Streams
  - Amazon SQS-Warteschlangen

Weitere Informationen finden Sie unter Was ist Amazon EventBridge? und Erste Schritte mit Amazon EventBridge im EventBridge Amazon-Benutzerhandbuch.

Eine EventBridge Regel erstellen (CLI)

Mit den Schritten im folgenden Beispiel wird eine EventBridge Regel erstellt, die ein Amazon SNS SNS-Thema veröffentlicht, wenn der angegebene `my-canary-name` Canary eine Ausführung `us-east-1` abschließt oder seinen Status ändert.

1. Erstellen Sie die -Regel.

```
aws events put-rule \
  --name TestRule \
  --region us-east-1 \
  --event-pattern "{\"source\": [\"aws.synthetics\"], \"detail\": {\"canary-name\":
 [\"my-canary-name\"]}}"
```

   Alle Eigenschaften, die Sie im Muster auslassen, werden ignoriert.

2. Fügen Sie das Thema als Regelziel hinzu.

   - Ersetzen Sie *topic-arn* durch den Amazon-Ressourcennamen (ARN) Ihres Amazon-SNS-Themas.

```
aws events put-targets \
  --rule TestRule \
  --targets "Id"="1","Arn"="topic-arn"
```

> **Note**
>
> Damit Amazon EventBridge Ihr Zielthema aufrufen kann, müssen Sie Ihrem Thema eine ressourcenbasierte Richtlinie hinzufügen. Weitere Informationen finden Sie unter Amazon SNS SNS-Berechtigungen im EventBridge Amazon-Benutzerhandbuch.

Weitere Informationen finden Sie unter Ereignisse und Ereignismuster EventBridge im EventBridge Amazon-Benutzerhandbuch.

Führen Sie Produkteinführungen und A/B-Experimente mit CloudWatch Eviently durch

Sie können Amazon CloudWatch Evidently verwenden, um neue Funktionen sicher zu validieren, indem Sie sie während der Einführung der Funktion einem bestimmten Prozentsatz Ihrer Nutzer zur Verfügung stellen. Sie können die Leistung des neuen Feature überwachen, um zu entscheiden, wann Sie den Traffic für Ihre Benutzer erhöhen möchten. Dadurch senken Sie Risiken und erkennen unbeabsichtigtes Verhalten noch bevor Sie das Feature vollständig einführen.

Sie können auch A/B-Experimente durchführen, um Features auf der Grundlage von Erkenntnissen und Daten zu gestalten. Ein Experiment kann bis zu fünf Varianten gleichzeitig testen. Evidently sammelt Versuchsdaten und analysiert sie mit statistischen Methoden. Es gibt auch klare Empfehlungen darüber, welche Varianten besser abschneiden. Sie können sowohl benutzerorientierte Funktionen als auch Backend-Funktionen testen.

Preisgestaltung für Evidently

Evidently belastet Ihr Konto basierend auf Evidently-Ereignissen und Evidently-Analyseeinheiten. Evidently-Ereignisse umfassen sowohl Datenereignisse wie Klicks und Seitenaufrufe als auch Zuweisungsereignisse, die bestimmen, welche Variante eines Features der Benutzer erhalten soll.

Evidently-Analyseeinheiten werden aus Evidently-Ereignissen generiert, basierend auf Regeln, die Sie in Evidently erstellt haben. Analyseeinheiten sind die Anzahl der Regelübereinstimmungen für Ereignisse. Beispielsweise könnte ein Klickereignis durch den Benutzer eine einzelne Evidently-Analyseeinheit erzeugen (in dem Fall die Anzahl der Klicks). Ein anderes Beispiel ist ein Kaufabwicklungsereignis durch den Benutzer, das zwei Evidently-Analyseeinheiten erzeugt: den Gesamtwert der Artikel im Warenkorb und deren Anzahl. Weitere Informationen zur Preisgestaltung finden Sie unter CloudWatch Amazon-Preise.

CloudWatch Offensichtlich ist es derzeit in den folgenden Regionen verfügbar:

- US East (Ohio)
- USA Ost (Nord-Virginia)
- USA West (Oregon)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)

- Europe (Frankfurt)
- Europa (Irland)
- Europa (Stockholm)

Themen

Zu verwendende IAM-Richtlinien für Evidently

Um CloudWatch Evidently vollständig verwalten zu können, müssen Sie als IAM-Benutzer oder als IAM-Rolle angemeldet sein und über die folgenden Berechtigungen verfügen:

- Die Richtlinie AmazonCloudWatchEvidentlyFullAccess
- Die Richtlinie ResourceGroupsandTagEditorReadOnlyAccess

Um ein Projekt erstellen zu können, das Bewertungsereignisse in Amazon S3 oder CloudWatch Logs speichert, benötigen Sie außerdem die folgenden Berechtigungen:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketPolicy",
                "s3:PutBucketPolicy",
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:DescribeResourcePolicies",
                "logs:PutResourcePolicy"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Zusätzliche Berechtigungen für die CloudWatch RUM-Integration

Wenn Sie außerdem Evidenly-Markteinführungen oder Experimente verwalten möchten, die in Amazon CloudWatch RUM integriert sind und RUM-Metriken zur Überwachung verwenden CloudWatch , benötigen Sie die AmazonCloudWatchFullAccessRUM-Richtlinie. Um eine IAM-Rolle zu erstellen, die dem CloudWatch RUM-Webclient die Erlaubnis erteilt, Daten an CloudWatch RUM zu senden, benötigen Sie die folgenden Berechtigungen:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:CreatePolicy",
                "iam:AttachRolePolicy"
            ],
            "Resource": [
```

```
                "arn:aws:iam::*:role/service-role/CloudWatchRUMEvidentlyRole-*",
                "arn:aws:iam::*:policy/service-role/CloudWatchRUMEvidentlyPolicy-*"
            ]
        }
    ]
}
```

Leseberechtigung für Evidently

Anderen Benutzern, die Evidenly-Daten einsehen, aber keine Evidenly-Ressourcen erstellen müssen, können Sie die Richtlinie gewähren. AmazonCloudWatchEvidentlyReadOnlyAccess

Erstellen von Projekten, Funktionen, Starts und Experimenten

Um mit CloudWatch Evidently zu beginnen, erstellen Sie entweder für einen Feature-Launch oder ein A/B-Experiment zunächst ein Projekt. Ein Projekt ist eine logische Gruppierung von Ressourcen. Innerhalb des Projekts erstellen Sie verschiedene Funktionen, die Sie dann testen oder launchen. Sie können ein Feature entweder erstellen, bevor Sie einen Start oder ein Experiment erstellen, oder gleichzeitig.

Themen

- Erstellen eines neuen Projekts
- Verwenden der clientseitigen Auswertung – unterstützt von AWS AppConfig
- Hinzufügen eines Features zu einem Projekt
- Fokussieren Ihrer Zielgruppe mithilfe von Segmenten
- Erstellen eines Starts
- Erstellen eines Experiments

Erstellen eines neuen Projekts

Gehen Sie wie folgt vor, um ein neues CloudWatch Evidently-Projekt einzurichten.

Um ein neues CloudWatch Evidently-Projekt zu erstellen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Evidently aus.
3. Wählen Sie Create project (Projekt erstellen) aus.

4. Geben Sie als Projektname einen Namen ein, anhand dessen dieses Projekt in der CloudWatch Evidently-Konsole identifiziert werden soll.

   Optional können Sie auch eine Projektbeschreibung eingeben.

5. Bei Evaluation event storage (Speicher für Auswertungsereignisse) wählen Sie aus, ob Sie die Evaluierungsereignisse speichern möchten, die Sie mit Evidently sammeln. Selbst wenn Sie diese Ereignisse nicht speichern, sammelt Evidently sie, um Metriken und andere Experimentdaten zu erstellen, die im Evidently Dashboard angezeigt werden. Weitere Informationen finden Sie unter Projekt-Datenspeicherung.

6. Wählen Sie für Use client-side evaluation (Clientseitige Auswertung verwenden) aus, ob Sie die clientseitige Auswertung für dieses Projekt aktivieren möchten. Bei der clientseitigen Evaluierung kann Ihre Anwendung den Benutzersitzungen lokal Varianten zuweisen, anstatt den Vorgang aufzurufen. EvaluateFeature Das mindert die Latenz- und Verfügbarkeitsrisiken, die mit einem API-Aufruf einhergehen. Weitere Informationen finden Sie unter Verwenden der clientseitigen Auswertung – unterstützt von AWS AppConfig.

   Um ein Projekt mit clientseitiger Auswertung zu erstellen, benötigen Sie die `evidently:ExportProjectAsConfiguration`-Berechtigung.

   Wenn Sie die clientseitige Auswertung aktivieren, gehen Sie auch folgendermaßen vor:

   a. Wählen Sie aus, ob Sie eine bestehende AWS AppConfig Anwendung verwenden oder eine neue erstellen möchten.

   b. Wählen Sie, ob Sie eine bestehende AWS AppConfig Umgebung verwenden oder eine neue erstellen möchten.

      Weitere Informationen zu Anwendungen und Umgebungen finden Sie AWS AppConfig unter So AWS AppConfig funktioniert es.

7. (Optional) Zum Hinzufügen von Tags zu diesem Projekt wählen Sie Tags, Add new tag (Neues Tag hinzufügen) aus.

   Geben Sie für Key (Schlüssel) einen Namen für das Tag ein. Sie können einen optionalen Wert für das Tag unter Value (Wert) hinzufügen.

   (Optional) Zum Hinzufügen eines weiteren Tags wählen Sie Add new tag (Neues Tag hinzufügen) erneut aus.

   Weitere Informationen finden Sie unter AWS Ressourcen kennzeichnen.

8. Wählen Sie Create project (Projekt erstellen) aus.

Verwenden der clientseitigen Auswertung – unterstützt von AWS AppConfig

Sie können die clientseitige Evaluation — powered by AWS AppConfig (clientseitige Evaluation) in einem Projekt verwenden, sodass Ihre Anwendung Benutzersitzungen lokal Varianten zuweisen kann, anstatt Varianten zuzuweisen, indem Sie den Vorgang aufrufen. EvaluateFeature Das mindert die Latenz- und Verfügbarkeitsrisiken, die mit einem API-Aufruf einhergehen.

Um die clientseitige Auswertung zu verwenden, fügen Sie die AWS AppConfig Lambda-Erweiterung als Ebene zu Ihren Lambda-Funktionen hinzu und konfigurieren Sie die Umgebungsvariablen. Die clientseitige Auswertung wird als Nebenprozess auf dem lokalen Host ausgeführt. Anschließend können Sie die Operationen und gegen aufrufen. EvaluationFeaturePutProjectEvent`localhost` Der clientseitige Auswertungsprozess behandelt die Variantenzuweisung, das Caching und die Datensynchronisierung. Weitere Informationen zu finden Sie AWS AppConfig unter So AWS AppConfig funktioniert es.

Bei der Integration mit AWS AppConfig geben Sie eine AWS AppConfig Anwendungs-ID und eine AWS AppConfig Umgebungs-ID für Evidently an. Sie können dieselbe Anwendungs-ID und Umgebungs-ID in allen Evidently-Projekten verwenden.

Wenn Sie ein Projekt mit aktivierter clientseitiger Evaluierung erstellen, erstellt Evidently ein AWS AppConfig Konfigurationsprofil für dieses Projekt. Das Konfigurationsprofil unterscheidet sich für jedes Projekt.

Zugriffskontrolle für clientseitige Auswertung

Die clientseitige Auswertung von Evidently verwendet einen anderen Zugriffskontrollmechanismus als der Rest von Evidently. Wir empfehlen Ihnen dringend, das zu beachten, damit Sie die richtigen Sicherheitsmaßnahmen implementieren können.

Mit Evidently können Sie IAM-Richtlinien erstellen, die Aktionen begrenzen, die ein Benutzer mit individuellen Ressourcen durchführen kann. Sie können beispielsweise eine Benutzerrolle erstellen, die einem Benutzer die Aktion verbietet. EvaluateFeature Weitere Informationen zu den Evidently-Aktionen, die mit IAM-Richtlinien gesteuert werden können, finden Sie unter Von Amazon CloudWatch Evidently definierte Aktionen.

Das clientseitige Auswertungsmodell ermöglicht lokale Auswertungen von Evidently-Funktionen, die Projektmetadaten verwenden. Ein Benutzer eines Projekts mit aktivierter clientseitiger

Evaluierung kann die EvaluateFeatureAPI für einen lokalen Host-Endpunkt aufrufen, und dieser API-Aufruf erreicht Evidenly nicht und wird auch nicht durch die IAM-Richtlinien des Evidenly-Dienstes authentifiziert. Dieser Aufruf ist auch dann erfolgreich, wenn der Benutzer nicht über die IAM-Berechtigung verfügt, die Aktion zu verwenden. EvaluateFeature Ein Benutzer benötigt jedoch weiterhin die PutProjectEventsErlaubnis, dass der Agent die Evaluierungsereignisse oder benutzerdefinierten Ereignisse zwischenspeichert und Daten asynchron nach Eviently auslagert.

Zusätzlich muss ein Benutzer die Berechtigung `evidently:ExportProjectAsConfiguration` haben, um ein Projekt erstellen zu können, das eine clientseitige Auswertung verwendet. Auf diese Weise können Sie den Zugriff auf EvaluateFeatureAktionen kontrollieren, die während der clientseitigen Auswertung aufgerufen werden.

Wenn Sie nicht vorsichtig sind, kann das Sicherheitsmodell der clientseitigen Auswertung die Richtlinien untergraben, die Sie für den Rest von Evidently festgelegt haben. Ein Benutzer mit der entsprechenden `evidently:ExportProjectAsConfiguration` Berechtigung kann ein Projekt mit aktivierter clientseitiger Evaluierung erstellen und dann die EvaluateFeatureAktion für die clientseitige Bewertung mit diesem Projekt verwenden, auch wenn ihm die Aktion in einer IAM-Richtlinie ausdrücklich verweigert wurde. EvaluateFeature

Erste Schritte mit Lambda

Evidently unterstützt derzeit die clientseitige Auswertung durch die Verwendung einer AWS Lambda -Umgebung. Entscheiden Sie zunächst, welche AWS AppConfig Anwendung und Umgebung Sie verwenden möchten. Wählen Sie eine vorhandene Anwendung und Umgebung aus oder erstellen Sie neue.

Mit den folgenden AWS AppConfig AWS CLI Beispielbefehlen werden eine Anwendung und eine Umgebung erstellt.

```
aws appconfig create-application --name YOUR_APP_NAME
```

```
aws appconfig create-environment --application-id YOUR_APP_ID --
name YOUR_ENVIRONMENT_NAME
```

Erstellen Sie als Nächstes mithilfe dieser AWS AppConfig Ressourcen ein Evidently-Projekt. Weitere Informationen finden Sie unter Erstellen eines neuen Projekts.

Die clientseitige Auswertung wird in Lambda durch die Verwendung einer Lambda-Ebene unterstützt. Dies ist eine öffentliche Ebene, die Teil einer öffentlichen AWS AppConfig Erweiterung istAWS-

`AppConfig-Extension`, die vom AWS AppConfig Service erstellt wurde. Weitere Informationen zu Lambda-Ebenen finden Sie unter Ebene.

Um die clientseitige Auswertung zu verwenden, müssen Sie diese Ebene zu Ihrer Lambda-Funktion hinzufügen sowie Berechtigungen und die Umgebungsvariablen konfigurieren.

So fügen Sie Ihrer Lambda-Funktion die Lambda-Ebene für die clientseitige Auswertung in Evidently hinzu und konfigurieren sie:

1. Erstellen Sie eine Lambda-Funktion, wenn das noch nicht geschehen ist.
2. Fügen Sie die clientseitige Auswertungsebene zu Ihrer Funktion hinzu. Sie können entweder seinen ARN angeben oder ihn aus der AWS Layer-Liste auswählen, falls Sie dies noch nicht getan haben. Weitere Informationen finden Sie unter Konfiguration von Funktionen für die Verwendung von Layern und Verfügbare Versionen der AWS AppConfig Lambda-Erweiterung.
3. Erstellen Sie eine IAM-Richtlinie EvidentlyAppConfigCachingAgentPolicymit dem folgenden Inhalt und fügen Sie sie der Ausführungsrolle der Funktion hinzu. Weitere Informationen finden Sie unter Lambda-Ausführungsrolle.

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Sid": "VisualEditor0",
               "Effect": "Allow",
               "Action": [
                   "appconfig:GetLatestConfiguration",
                   "appconfig:StartConfigurationSession",
                   "evidently:PutProjectEvents"
               ],
               "Resource": "*"
           }
       ]
   }
   ```

4. Fügen Sie die erforderliche Umgebungsvariable `AWS_APPCONFIG_EXTENSION_EVIDENTLY_CONFIGURATIONS` zu Ihrer Lambda-Funktion hinzu. Diese Umgebungsvariable spezifiziert die Zuordnung zwischen dem Evidenly-Projekt und den AWS AppConfig Ressourcen.

Wenn Sie diese Funktion für ein Evidently-Projekt verwenden, setzen Sie den Wert der Umgebungsvariablen auf: `applications/APP_ID/environments/ENVIRONMENT_ID/configurations/PROJECT_NAME`

Wenn Sie diese Funktion für mehrere Evidently-Projekte verwenden, trennen Sie die Werte durch ein Komma, wie im folgenden Beispiel gezeigt: `applications/APP_ID_1/environments/ENVIRONMENT_ID_1/configurations/PROJECT_NAME_1, applications/APP_ID_2/environments/ENVIRONMENT_ID_2/configurations/PROJECT_NAME_2`

5. (Optional) Legen Sie andere Umgebungsvariablen fest. Weitere Informationen finden Sie unter Konfiguration der AWS AppConfig Lambda-Erweiterung.

6. In Ihrer Anwendung erhalten Sie Evidently-Auswertungen lokal, indem Sie `EvaluateFeature` an `localhost` senden.

   Python-Beispiel:

```
import boto3
from botocore.config import Config

def lambda_handler(event, context):
    local_client = boto3.client(
        'evidently',
        endpoint_url="http://localhost:2772",
        config=Config(inject_host_prefix=False)
    )
    response = local_client.evaluate_feature(
        project=event['project'],
        feature=event['feature'],
        entityId=event['entityId']
    )
    print(response)
```

   Node.js-Beispiel:

```
const AWS = require('aws-sdk');
const evidently = new AWS.Evidently({
    region: "us-west-2",
    endpoint: "http://localhost:2772",
    hostPrefixEnabled: false
```

```
});

exports.handler = async (event) => {

    const evaluation = await evidently.evaluateFeature({
        project: 'John_ETCProject_Aug2022',
        feature: 'Feature_IceCreamFlavors',
        entityId: 'John'
    }).promise()

    console.log(evaluation)
    const response = {
        statusCode: 200,
        body: evaluation,
    };
    return response;
};
```

Kotlin-Beispiel:

```
String localhostEndpoint = "http://localhost:2772/"
public AmazonCloudWatchEvidentlyClient getEvidentlyLocalClient() {
    return AmazonCloudWatchEvidentlyClientBuilder.standard()

 .withEndpointConfiguration(AwsClientBuilder.EndpointConfiguration(localhostEndpoint,
 region))

 .withClientConfiguration(ClientConfiguration().withDisableHostPrefixInjection(true))
        .withCredentials(credentialsProvider)
        .build();
}

AmazonCloudWatchEvidentlyClient evidently = getEvidentlyLocalClient();

// EvaluateFeature via local client.
EvaluateFeatureRequest evaluateFeatureRequest = new
 EvaluateFeatureRequest().builder()
   .withProject(${YOUR_PROJECT}) //Required.
   .withFeature(${YOUR_FEATURE}) //Required.
   .withEntityId(${YOUR_ENTITY_ID}) //Required.
   .withEvaluationContext(${YOUR_EVAL_CONTEXT}) //Optional: a JSON object of
 attributes that you can optionally pass in as part of the evaluation event sent to
 Evidently.
```

```
   .build();

EvaluateFeatureResponse evaluateFeatureResponse =
 evidently.evaluateFeature(evaluateFeatureRequest);

// PutProjectEvents via local client.
PutProjectEventsRequest putProjectEventsRequest = new
 PutProjectEventsRequest().builder()
    .withData(${YOUR_DATA})
    .withTimeStamp(${YOUR_TIMESTAMP})
    .withType(${YOUR_TYPE})
    .build();

PutProjectEvents putProjectEventsResponse =
 evidently.putProjectEvents(putProjectEventsRequest);
```

Konfigurieren, wie oft der Client Daten an Evidently sendet

Um festzulegen, wie oft die clientseitige Auswertung Daten an Evidently sendet, können Sie optional zwei Umgebungsvariablen konfigurieren.

- `AWS_APPCONFIG_EXTENSION_EVIDENTLY_EVENT_BATCH_SIZE` gibt die Anzahl der Ereignisse pro Projekt an, die gebündelt werden, bevor sie an Evidently gesendet werden. Gültig sind Ganzzahlwerte zwischen 1 und 50, und der Standardwert ist 40.
- `AWS_APPCONFIG_EXTENSION_EVIDENTLY_BATCH_COLLECTION_DURATION` gibt die Dauer in Sekunden an, die auf Ereignisse gewartet werden soll, bevor sie an Evidently gesendet werden. Der Standardwert ist 30.

Fehlerbehebung

Verwenden Sie die folgenden Informationen, um Probleme bei der Verwendung von CloudWatch Evidently mit clientseitiger Evaluierung — powered by zu beheben. AWS AppConfig

Beim Aufrufen des EvaluateFeature Vorgangs ist ein Fehler aufgetreten (BadRequestException): Die HTTP-Methode wird für den angegebenen Pfad nicht unterstützt

Ihre Umgebungsvariablen sind möglicherweise falsch konfiguriert. Beispiel: Sie haben vielleicht `EVIDENTLY_CONFIGURATIONS` als Umgebungsvariablenname anstatt `AWS_APPCONFIG_EXTENSION_EVIDENTLY_CONFIGURATIONS` verwendet.

ResourceNotFoundException: Die Bereitstellung wurde nicht gefunden

Ihre Aktualisierung der Projektmetadaten wurde nicht für AWS AppConfig bereitgestellt. Suchen Sie nach einer aktiven Bereitstellung in der AWS AppConfig Umgebung, die Sie für die clientseitige Evaluierung verwendet haben.

ValidationException: Offensichtlich keine Konfiguration für das Projekt

Ihre `AWS_APPCONFIG_EXTENSION_EVIDENTLY_CONFIGURATIONS`-Umgebungsvariable ist möglicherweise mit dem falschen Projektnamen konfiguriert.

Hinzufügen eines Features zu einem Projekt

Eine Funktion in CloudWatch Evidently stellt eine Funktion dar, die Sie starten oder von der Sie Varianten testen möchten.

Bevor Sie ein Feature hinzufügen können, müssen Sie ein Projekt erstellen. Weitere Informationen finden Sie unter Erstellen eines neuen Projekts.

Ein Feature zu einem Projekt hinzufügen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Evidently aus.
3. Wählen Sie den Namen des Projekts aus.
4. Wählen Sie Add feature (Funktion hinzufügen) aus.
5. Geben Sie bei Feature name (Funktionsname) einen Namen ein, mit dem diese Funktion im Projekt kenntlich gemacht werden soll.

   Optional können Sie auch eine Feature-Beschreibung eingeben.
6. Bei den Feature variations (Funktionsvarianten) können Sie den Variation Type (Typ der Variante) wählen: Boolean, Long, Double oder String. Weitere Informationen finden Sie unter Typen von Varianten.
7. Fügen Sie bis zu fünf Varianten für Ihr Feature hinzu. Der Value (Wert) einer jeden Variante muss ihrem Variation type (Typ der Variante) entsprechen.

   Legen Sie eine der Varianten als Standardvariante fest. Sie ist das Muster, mit dem alle anderen Varianten verglichen werden, und sollte die Variante sein, die Ihren Benutzern aktuell zur Verfügung gestellt wird. Es ist auch die Variante, die Benutzern zu Verfügung gestellt wird, die nicht zu einem Start oder Experiment für dieses Feature hinzugefügt werden.

8. Wählen Sie Sample code (Beispiel-Code) aus. Das Codebeispiel zeigt, was Sie Ihrer Anwendung hinzufügen müssen, um die Varianten einzurichten und ihnen Benutzersitzungen zuzuweisen. Sie können für den Code zwischen JavaScript Java und Python wählen.

   Der Programmcode muss der Anwendung nicht sofort hinzugefügt werden, allerdings bevor Sie einen Start oder ein Experiment beginnen.

   Weitere Informationen finden Sie unter Hinzufügen von Programmcode zur Anwendung.

9. (Optional) Um anzugeben, dass bestimmte Benutzer immer eine bestimmte Variante sehen, wählen Sie Overrides (Überschreibungen) und dann Add override (Überschreibung hinzufügen) aus. Geben Sie dann einen Benutzer an, indem Sie seine Benutzer-ID, Konto-ID oder eine andere Kennung in Identifier (ID) eingeben und bestimmen, welche Variante er sehen soll.

   Dies kann für Mitglieder Ihres eigenen Testteams oder andere interne Benutzer nützlich sein, wenn Sie sicherstellen möchten, dass sie eine bestimmte Variante sehen. Die Sitzungen von Benutzern, denen Überschreibungen zugewiesen wurden, tragen nicht zu Start- oder Experimentmetriken bei.

   Sie können dies für bis zu 20 Benutzer wiederholen, indem Sie erneut Add Override wählen.

10. (Optional) Zum Hinzufügen von Tags zu diesem Feature wählen Sie Tags, Add new tag (Neues Tag hinzufügen) aus.

    Geben Sie für Key (Schlüssel) einen Namen für das Tag ein. Sie können einen optionalen Wert für das Tag unter Value (Wert) hinzufügen.

    (Optional) Zum Hinzufügen eines weiteren Tags wählen Sie Add new tag (Neues Tag hinzufügen) erneut aus.

    Weitere Informationen finden Sie unter AWS Ressourcen taggen.

11. Wählen Sie Add feature (Funktion hinzufügen) aus.

Typen von Varianten

Wenn Sie ein Feature erstellen und die Varianten definieren, geben Sie auch den Typ der Variante an. Die möglichen Typen sind:

- Boolesch
- Long (Ganzzahl)

- Double (Gleitkommazahl mit doppelter Genauigkeit)
- String (Zeichenfolge)

Der Typ der Variante bestimmt, wie die verschiedenen Varianten in Ihrem Code unterschieden werden. Sie können den Variationstyp verwenden, um die Implementierung von CloudWatch Evidently zu vereinfachen und auch die Änderung der Funktionen in Ihren Produkteinführungen und Experimenten zu vereinfachen.

Wenn Sie beispielsweise ein Feature mit dem Variantentyp „long“ definieren, können die Ganzzahlen, die Sie zur Unterscheidung der Varianten angeben, Zahlen sein, die direkt in Ihren Code übergeben werden. Ein Beispiel könnte das Testen der Pixelgröße einer Schaltfläche sein. Die Werte für die Variantentypen können die Anzahl der Pixel sein, die in jeder Variante verwendet werden. Die Codes für die einzelnen Varianten können den Wert des Variantentyps lesen und diesen für die Schaltflächengröße verwenden. Um eine neue Schaltflächengröße zu testen, können Sie die für den Wert der Variante verwendete Zahl ändern, ohne weitere Änderungen am Code vorzunehmen.

Wenn Sie die Werte für Ihre Variationstypen innerhalb einer Funktion festlegen, sollten Sie vermeiden, dieselben Werte mehreren Varianten zuzuweisen, es sei denn, Sie möchten A/A-Tests durchführen, um CloudWatch Evidently zunächst auszuprobieren, oder Sie haben andere Gründe dafür.

Evidently hat keine native Unterstützung für JSON als Typ, aber Sie können JSON im Variantentyp „String“ übergeben und diesen JSON in Ihrem Code analysieren.

Fokussieren Ihrer Zielgruppe mithilfe von Segmenten

Sie können Segmente für Zielgruppen definieren und sie in Ihren Starts und Experimenten verwenden. Ein Segment ist ein Teil Ihrer Zielgruppe, die mindestens ein Merkmal gemeinsam hat. Beispiele wären etwa Benutzer des Chrome-Browsers, Benutzer in Europa oder Benutzer des Firefox-Browsers in Europa, die noch weitere von Ihrer Anwendung erfasste Kriterien erfüllen (beispielsweise Alter).

Die Verwendung eines Segments in einem Experiment schränkt dieses Experiment so ein, dass nur die Benutzer ausgewertet werden, die die Kriterien des Segments erfüllen. Wenn Sie eines oder mehrere Segmente in einem Start verwenden, können Sie verschiedene Datenverkehrsaufteilungen für die verschiedenen Zielgruppensegmente definieren.

Syntax für Segmentregelmuster

Um ein Segment zu erstellen, definieren Sie ein Segmentregelmuster. Legen Sie die Attribute fest, auf deren Grundlage ausgewertet werden soll, ob eine Benutzersitzung zu dem Segment gehört. Das Muster, das Sie erstellen, wird mit dem Wert von `evaluationContext` verglichen, den Evidently in einer Benutzersitzung findet. Weitere Informationen finden Sie unter Verwenden EvaluateFeature.

Geben Sie zum Erstellen eines Segmentregelmusters die Felder an, denen das Muster entsprechen soll. Sie können in Ihrem Muster auch Logik wie `And`, `Or`, `Not` und `Exists` verwenden.

Damit ein Auswertungskontext (`evaluationContext`) einem Muster entspricht, muss der Auswertungskontext (`evaluationContext`) allen Teilen des Regelmusters entsprechen. Die Felder im Auswertungskontext (`evaluationContext`), die nicht im Regelmuster enthalten sind, werden von Evidently ignoriert.

Für die von Regelmustern abgeglichenen Werte gelten JSON-Regeln. Sie können in Anführungszeichen (") gesetzte Zeichenfolgen sowie Zahlen und die Schlüsselwörter `true`, `false` und `null` verwenden.

Für Zeichenketten verwendet Evidently den exakten character-by-character Abgleich ohne Umschaltung der Groß- und Kleinschreibung oder eine andere Normalisierung von Zeichenketten. Daher muss bei Regelübereinstimmungen die Groß-/Kleinschreibung beachtet werden. Beispiel: Wenn ihr `evaluationContext` ein `browser`-Attribut beinhaltet, aber Ihr Regelmuster auf `Browser` prüft, gibt es keine Übereinstimmung.

Für Zahlen verwendet Evidently eine Zeichenfolgendarstellung. 300, 300,0 und 3,0e2 werden z. B. nicht gleich behandelt.

Wenn Sie Regelmuster für den Abgleich von `evaluationContext` schreiben, können Sie die API `TestSegmentPattern` oder den CLI-Befehl `test-segment-pattern` verwenden, um zu testen, ob Ihr Muster dem korrekten JSON-Code entspricht. Weitere Informationen finden Sie unter. TestSegmentPattern

Die folgende Zusammenfassung zeigt alle Vergleichsoperatoren an, die in Evidently-Segmentmustern verfügbar sind:

Vergleich	Beispiel	Regelsyntax
Null	UserID is null	`{`

Vergleich	Beispiel	Regelsyntax

		<pre>"UserID": [null]
		}</pre>

Leer	LastName ist leer	<pre>{
		"LastName": [""]
		}</pre>

Gleichheitszeichen	Browser ist „Chrome“	<pre>{
		"Browser":
		["Chrome"]
		}</pre>

And	Land ist „Frankreich“ und Gerät ist „Mobil“	<pre>{
		"Country":
		["France"], "Device":
		["Mobile"]
		}</pre>

Oder (mehrere Werte eines einzelnen Attributs)	Browser ist „Chrome“ oder „Firefox“	<pre>{
		"Browser":
		["Chrome", "Firefox"]
		}</pre>

Oder (verschiedene Attribute)	Browser ist „Safari“ oder Gerät ist „Tablet“	<pre>{
		"$or": [
		{"Browser":
		["Safari"]},
		{"Device": ["Tablet"
		}]
		]
		}</pre>

Vergleich	Beispiel	Regelsyntax
Nicht	Browser ist alles andere als „Safari“	`{ "Browser": [ { "anything-but": [ "Safari" ] } ] }`
Numerisch (ist gleich)	Price is 100	`{ "Price": [ { "numeric": [ "=", 100 ] } ] }`
Numerisch (Bereich)	Price is more than 10, and less than or equal to 20	`{ "Price": [ { "numeric": [ ">", 10, "<=", 20 ] } ] }`
Vorhanden	Altersfeld ist vorhanden	`{ "Age": [ { "exists": true } ] }`
Nicht vorhanden	Altersfeld ist nicht vorhanden	`{ "Age": [ { "exists": false } ] }`
Beginnt mit einem Präfix	Region ist in den USA	`{ "Region": [ {"prefix": "us-" } ] }`

Vergleich	Beispiel	Regelsyntax
Endet mit einem Suffix	Standort hat das Suffix „West“	`{ "Region": [ {"suffix": "West" } ] }`

Beispiele für Segmentregeln

Bei den folgenden Beispielen wird jeweils vorausgesetzt, dass Sie Werte für `evaluationContext` mit den gleichen Feldbezeichnungen und Werten übergeben, die Sie auch in Ihren Regelmustern verwenden.

Im folgenden Beispiel wird überprüft, ob `Browser` „Chrome“ oder „Firefox“ und ob `Location` „US-West“ ist.

```
{
  "Browser": ["Chrome", "Firefox"],
  "Location": ["US-West"]
}
```

Im folgenden Beispiel wird überprüft, ob `Browser` ein beliebiger Browser außer Chrome ist, ob `Location` mit US beginnt und ob ein Feld vom Typ `Age` vorhanden ist:

```
{
  "Browser": [ {"anything-but": ["Chrome"]}],
  "Location": [{"prefix": "US"}],
  "Age": [{"exists": true}]
}
```

Im folgenden Beispiel wird überprüft, ob `Location` „Japan“ ist und ob entweder `Browser` „Safari“ oder `Device` „Tablet“ ist.

```
{
 "Location": ["Japan"],
 "$or": [
   {"Browser": ["Safari"]},
```

```
    {"Device": ["Tablet"]}
  ]
}
```

Erstellen eines Segments

Nachdem Sie ein Segment erstellt haben, können Sie es bei jedem Start oder Experiment in einem beliebigen Projekt verwenden.

So erstellen Sie ein Segment

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Evidently aus.
3. Wählen Sie die Registerkarte Segments (Segmente) aus.
4. Wählen Sie Create segment (Segment erstellen) aus.
5. Geben Sie unter Segment name (Segmentname) einen Namen zur Identifizierung dieses Segments ein.

   Sie können optional auch eine Beschreibung hinzufügen.
6. Geben Sie unter Segment pattern (Segmentmuster) einen JSON-Block zum Definieren des Regelmusters ein. Weitere Informationen zur Syntax von Regelmustern finden Sie unter Syntax für Segmentregelmuster.

Erstellen eines Starts

Um ein neues Feature verfügbar zu machen oder eine bestimmten Prozentsatz Ihrer Benutzer einzustellen, erstellen Sie einen Start. Sie können dann wichtige Metriken wie Seitenladezeiten und Konvertierungen überwachen, bevor Sie das Feature für alle Ihre Benutzer bereitstellen.

Bevor Sie einen Start hinzufügen können, müssen Sie ein Projekt erstellen. Weitere Informationen finden Sie unter Erstellen eines neuen Projekts.

Wenn Sie einen Start hinzufügen, können Sie ein Feature verwenden, die Sie bereits erstellt haben, oder bei der Erstellung des Starts eine neue Funktion erstellen.

Einen Start zu einem Projekt hinzufügen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

2. Wählen Sie im Navigationsbereich Application Signals, Evidently aus.
3. Wählen Sie die Schaltfläche neben dem Namen des Projekts und wählen Sie Project actions (Projektaktionen) und Create launch (Start erstellen) aus.
4. Geben Sie bei Launch name (Startname) einen Namen ein, der zur Kenntlichmachung dieses Features in diesem Projekt verwendet werden soll.

   Optional können Sie auch eine Beschreibung hinzufügen.
5. Wählen Sie entweder Select from existing features (Aus bestehenden Funktionen auswählen) oder Add new feature (Neue Funktion hinzufügen) aus.

   Wenn Sie eine vorhandene Funktion verwenden, wählen Sie sie unter Feature name (Funktionsname) aus.

   Wenn Sie die Option Add new feature (Neue Funktion hinzufügen) auswählen, gehen Sie wie folgt vor:

   a. Geben Sie bei Feature name (Funktionsname) einen Namen ein, mit dem diese Funktion im Projekt kenntlich gemacht werden soll. Optional können Sie auch eine Beschreibung hinzufügen.
   b. Bei den Feature variations (Funktionsvarianten) können Sie den Variation Type (Typ der Variante) wählen: Boolean, Long, Double oder String. Weitere Informationen finden Sie unter Typen von Varianten.
   c. Fügen Sie bis zu fünf Varianten für Ihr Feature hinzu. Der Value (Wert) einer jeden Variante muss ihrem Variation type (Typ der Variante) entsprechen.

      Legen Sie eine der Varianten als Standardvariante fest. Sie ist das Muster, mit dem alle anderen Varianten verglichen werden, und sollte die Variante sein, die Ihren Benutzern aktuell zur Verfügung gestellt wird. Wenn Sie ein Experiment beenden, wird diese Standardvariante dann allen Benutzern bereitgestellt.
   d. Wählen Sie Sample code (Beispiel-Code) aus. Das Codebeispiel zeigt, was Sie Ihrer Anwendung hinzufügen müssen, um die Varianten einzurichten und ihnen Benutzersitzungen zuzuweisen. Sie können für den Code zwischen JavaScript Java und Python wählen.

      Der Programmcode muss der Anwendung nicht sofort hinzugefügt werden, allerdings bevor Sie den Start beginnen.

      Weitere Informationen finden Sie unter Hinzufügen von Programmcode zur Anwendung.

6. Bei der Startkonfiguration (Launch configuration) wählen Sie, ob der Start sofort oder später gestartet werden soll.
7. (Optional) Wenn Sie für von Ihnen definierte Zielgruppensegmente unterschiedliche Datenverkehrsaufteilungen festlegen möchten, wählen Sie anstelle der Datenverkehrsaufteilung, die Sie für Ihre allgemeine Zielgruppe verwenden, die Option Add Segment Overrides (Segmentüberschreibungen hinzufügen) aus.

   Wählen Sie unter Segment Overrides (Segmentüberschreibungen) ein Segment aus und definieren Sie die gewünschte Datenverkehrsaufteilung für dieses Segment.

   Sie können optional weitere Segmente definieren, für die Datenverkehrsaufteilungen definiert werden sollen, indem Sie Add Segment Override (Segmentüberschreibung hinzufügen) auswählen. Für einen Start können bis zu sechs Segmentüberschreibungen vorhanden sein.

   Weitere Informationen finden Sie unter Fokussieren Ihrer Zielgruppe mithilfe von Segmenten.
8. Wählen Sie unter Traffic configuration (Datenverkehrskonfiguration) den prozentualen Anteil des Datenverkehrs aus, der den einzelnen Varianten für die allgemeine Zielgruppe zugewiesen werden soll, die nicht den Segmentüberschreibungen entsprechen. Sie können auch festlegen, dass Varianten von der Bereitstellung an Benutzer ausgeschlossen werden.

   Die Traffic summary (Datenverkehrsübersicht) zeigt an, wie viel von Ihrem Gesamt-Datenverkehr für diesen Start verfügbar ist.
9. Wenn Sie den Start später launchen lassen möchten, können Sie dem Start mehrere Schritte hinzufügen. Jeder Schritt kann verschiedene Prozentsätze zum Bereitstellen der Varianten verwenden. Wählen Sie dazu die Option Add another step (Weiteren Schritt hinzufügen) aus und geben Sie dann den Zeitplan und die Datenverkehr-Prozentanteile für den nächsten Schritt an. Sie können bis zu fünf Schritte in einen Start einbinden.
10. Wenn Sie die Leistung des Features während des Starts mit Metriken verfolgen möchten, wählen Sie Metrics (Metriken) und Add metric (Metrik hinzufügen) aus. Sie können entweder CloudWatch RUM-Metriken oder benutzerdefinierte Metriken verwenden.

    Um eine benutzerdefinierte Metrik zu verwenden, können Sie die Metrik hier mithilfe einer EventBridge Amazon-Regel erstellen. Gehen Sie wie folgt vor, um eine benutzerdefinierte Metrik zu erstellen:

    - Wählen Sie Eigene Metriken aus und geben Sie einen Namen für die Metrik ein.

- Geben Sie unter Metric rule (Metrikregel) bei Entity ID (Entitäts-ID) die Identifikationsmethode der Entität ein. Dies kann ein Benutzer oder eine Sitzung sein, die eine Aktion ausführen, die bewirkt, dass ein Metrikwert aufgezeichnet wird. Ein Beispiel ist `userDetails.userID`.
- Geben Sie bei Value key (Werteschlüssel), den Wert ein, der verfolgt werden soll, um die Metrik zu erzeugen.
- Geben Sie optional einen Namen für die Einheiten für die Metrik ein. Dieser Einheitenname dient nur zur Anzeige in Diagrammen in der Evidently-Konsole.

Wenn Sie diese Felder eingeben, zeigt das Feld Beispiele für die Codierung der EventBridge Regel zur Erstellung der Metrik. Weitere Informationen zu EventBridge finden Sie unter Was ist Amazon EventBridge?

Um RUM-Metriken verwenden zu können, müssen Sie bereits eine RUM-App-Überwachung für Ihre Anwendung eingerichtet haben. Weitere Informationen finden Sie unter Richten Sie eine Anwendung zur Verwendung von CloudWatch RUM ein.

> **Note**
>
> Wenn Sie RUM-Metriken verwenden und die App-Überwachung nicht für die Prüfung von 100 % der Benutzersitzungen konfiguriert ist, senden nicht alle Benutzersitzungen, die am Start teilnehmen, Metriken an Evidently. Um sicherzustellen, dass die Startmetriken korrekt sind, empfehlen wir, dass die App-Überwachung 100 % der Benutzersitzungen für das Sampling verwendet.

11. (Optional) Wenn Sie mindestens eine Metrik für den Start erstellen, können Sie diesem Start einen vorhandenen CloudWatch Alarm zuordnen. Wählen Sie dazu CloudWatch Alarme zuordnen aus.

    Wenn Sie einen Alarm mit einem Start verknüpfen, CloudWatch müssen Sie dem Alarm natürlich Tags mit dem Projektnamen und dem Startnamen hinzufügen. Auf diese Weise kann CloudWatch Evidently die richtigen Alarme in den Startinformationen in der Konsole anzeigen.

    Um zu bestätigen, dass CloudWatch Evidently diese Tags hinzufügt, wählen Sie Allow Eviently, um die unten angegebene Alarm-Ressource mit dieser Startressource zu kennzeichnen. Wählen Sie dann Associate Alarm (Alarm zuordnen) aus und geben Sie den Alarmnamen ein.

Informationen zum Erstellen von CloudWatch Alarmen finden Sie unter. CloudWatch Amazon-Alarme verwenden

12. (Optional) Zum Hinzufügen von Tags zu diesem Start wählen Sie Tags, Add new tag (Neues Tag hinzufügen) aus.

    Geben Sie für Key (Schlüssel) einen Namen für das Tag ein. Sie können einen optionalen Wert für das Tag unter Value (Wert) hinzufügen.

    (Optional) Zum Hinzufügen eines weiteren Tags wählen Sie Add new tag (Neues Tag hinzufügen) erneut aus.

    Weitere Informationen finden Sie unter AWS Ressourcen taggen.
13. Wählen Sie Create launch (Start erstellen) aus.

Erstellen eines Experiments

Nutzen Sie Experimente, um verschiedene Versionen eines Features oder einer Website zu testen und Daten aus echten Benutzersitzungen zu sammeln. Auf diese Weise können Sie basierend auf gesammelten Daten Entscheidungen für Ihre Anwendung treffen.

Bevor Sie ein Experiment hinzufügen können, müssen Sie ein Projekt erstellen. Weitere Informationen finden Sie unter Erstellen eines neuen Projekts.

Wenn Sie ein Experiment hinzufügen, können Sie ein Feature verwenden, die Sie bereits erstellt haben, oder mit dem Experiment gleich eine neue Funktion erstellen.

Einem Projekt ein Experiment hinzufügen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Evidently aus.
3. Wählen Sie die Schaltfläche neben dem Namen des Projekts und wählen Sie Project actions (Aktionen für das Projekt) und dann Create experiment (Experiment erstellen) aus.
4. Geben Sie bei Experiment name (Experimentname) einen Namen ein, der zur Kenntlichmachung dieses Features in diesem Projekt verwendet werden soll.

    Optional können Sie auch eine Beschreibung hinzufügen.
5. Wählen Sie entweder Select from existing features (Aus bestehenden Funktionen auswählen) oder Add new feature (Neue Funktion hinzufügen) aus.

Wenn Sie eine vorhandene Funktion verwenden, wählen Sie sie unter Feature name (Funktionsname) aus.

Wenn Sie die Option Add new feature (Neue Funktion hinzufügen) auswählen, gehen Sie wie folgt vor:

a. Geben Sie bei Feature name (Funktionsname) einen Namen ein, mit dem diese Funktion im Projekt kenntlich gemacht werden soll. Optional können Sie auch eine Beschreibung eingeben.

b. Bei den Feature variations (Funktionsvarianten) können Sie den Variation Type (Typ der Variante) wählen: Boolean, Long, Double oder String. Der Typ definiert, welcher Werttyp für die einzelnen Varianten verwendet wird. Weitere Informationen finden Sie unter Typen von Varianten.

c. Fügen Sie bis zu fünf Varianten für Ihr Feature hinzu. Der Value (Wert) einer jeden Variante muss ihrem Variation type (Typ der Variante) entsprechen.

   Legen Sie eine der Varianten als Standardvariante fest. Sie ist das Muster, mit dem alle anderen Varianten verglichen werden, und sollte die Variante sein, die Ihren Benutzern aktuell zur Verfügung gestellt wird. Wenn Sie ein Experiment beenden, das dieses Feature verwendet, wird die Standardvariante dem Teil der Benutzer bereitgestellt, die zuvor das Experiment genutzt hatten.

d. Wählen Sie Sample code (Beispiel-Code) aus. Das Codebeispiel zeigt, was Sie Ihrer Anwendung hinzufügen müssen, um die Varianten einzurichten und ihnen Benutzersitzungen zuzuweisen. Sie können für den Code zwischen JavaScript Java und Python wählen.

   Sie müssen den Programmcode nicht gleich zu Ihrer Anwendung hinzufügen, aber noch bevor Sie mit dem Experiment beginnen. Weitere Informationen finden Sie unter Hinzufügen von Programmcode zur Anwendung.

6. Wählen Sie unter Audience (Zielgruppe) optional ein von Ihnen erstelltes Segment aus, wenn dieses Experiment nur für die Benutzer gelten soll, die diesem Segment entsprechen. Weitere Informationen zu Segmenten finden Sie unter Fokussieren Ihrer Zielgruppe mithilfe von Segmenten.

7. Geben Sie unter Traffic split for the experiment (Datenverkehrsaufteilung für das Experiment) den prozentualen Anteil der ausgewählten Zielgruppe an, deren Sitzungen im Experiment

verwendet werden. Weisen Sie dann den Datenverkehr für die verschiedenen Varianten zu, die das Experiment verwendet.

Wenn ein Start und ein Experiment gleichzeitig für dasselbe Feature laufen, wird die Zielgruppe zuerst zum Start geleitet. Dann wird der für den Start angegebene Anteil des Datenverkehrs von der gesamten Zielgruppe übernommen. Danach ist der Anteil, den Sie hier angeben, der Anteil der verbleibenden Zielgruppe, die für das Experiment verwendet wird. Der übrige Datenverkehr danach wird für die Standardvariante bereitgestellt.

8. Wählen Sie bei Metrics (Metriken) die Metriken aus, die zur Bewertung der Varianten während des Experiments verwendet werden sollen. Sie müssen für die Auswertung mindestens eine Metrik verwenden.

   a. Wählen Sie unter Metric Source aus, ob Sie CloudWatch RUM-Metriken oder benutzerdefinierte Metriken verwenden möchten.

   b. Geben Sie einen Namen für die Rolle ein. Wählen Sie bei Ziel die Option Increase (Erhöhen) aus, wenn Sie möchten, dass ein höherer Wert für die Metrik auf eine bessere Variante hinweist. Wählen Sie Decrease (Verringern) aus, wenn Sie möchten, dass ein niedrigerer Wert für die Metrik auf eine bessere Variante hinweist.

   c. Wenn Sie eine benutzerdefinierte Metrik verwenden, können Sie die Metrik hier mithilfe einer EventBridge Amazon-Regel erstellen. Gehen Sie wie folgt vor, um eine benutzerdefinierte Metrik zu erstellen:

      - Geben Sie unter Metric rule (Metrikregel) bei Entity ID (Entitäts-ID) an, wie die die Entität identifiziert werden soll. Dies kann ein Benutzer oder eine Sitzung sein, der/die eine Aktion ausführt, die die Aufzeichnung eines Metrikwerts auslöst. Ein Beispiel ist `userDetails.userID`.
      - Geben Sie bei Value key (Werteschlüssel), den Wert ein, der verfolgt werden soll, um die Metrik zu erzeugen.
      - Geben Sie optional einen Namen für die Einheiten für die Metrik ein. Dieser Einheitenname dient nur zur Anzeige in Diagrammen in der Evidently-Konsole.

      Sie können RUM-Metriken nur verwenden, wenn Sie RUM zur Überwachung dieser Anwendung eingerichtet haben. Weitere Informationen finden Sie unter Verwenden Sie CloudWatch RUM.

> **Note**
>
> Wenn Sie RUM-Metriken verwenden und die App-Überwachung nicht für die Probe von 100 % der Benutzersitzungen konfiguriert ist, senden nicht alle Benutzersitzungen im Experiment Metriken an Evidently. Um sicherzustellen, dass die Metriken des Experiments korrekt sind, empfehlen wir, dass die App-Überwachung 100 % der Benutzersitzungen für das Sampling verwendet.

d. (Optional) Um weitere zu bewertende Metriken hinzuzufügen, wählen Sie Add metric (Metrik hinzufügen) aus. Sie können während des Experiments bis zu drei Metriken auswerten.

9. (Optional) Um CloudWatch Alarme für dieses Experiment zu erstellen, wählen Sie CloudWatchAlarme. Die Alarme können überwachen, ob die Ergebnisse der einzelnen Varianten und der Standardvariante größer als ein von Ihnen festgelegter Schwellenwert ist. Wenn die Leistung einer Variante schlechter als die Standardvariante und der Unterschied größer als der Schwellenwert ist, wechselt sie in den Alarmzustand und Sie werden benachrichtigt.

   Wenn Sie hier einen Alarm erstellen, wird für jede Variante ein Alarm erzeugt, der nicht die Standardvariante ist.

   Wenn Sie einen neuen Alarm erstellen, geben Sie Folgendes an:

   - Wählen Sie für Metric name (Metrikname) die für den Alarm zu verwendende Experimentmetrik aus.
   - Wählen Sie bei Alarm condition (Alarmbedingung) aus, bei welcher Bedingung der Alarmzustand eintritt, wenn die Metrikwerte der Variante mit den Metrikwerten der Standardvariante verglichen werden. Wählen Sie beispielsweise Greater (größer) oder Greater/Equal (größer/gleich) aus, wenn höhere Werte anzeigen sollen, dass die Variante schlecht funktioniert. Das wäre sinnvoll, wenn die Metrik beispielsweise die Ladezeit der Seite misst.
   - Geben Sie eine Zahl für den Schwellenwert ein, d. h. ab welcher prozentualen Abweichung der Leistung der Alarm in den Zustand `ALARM` wechselt.
   - Wählen Sie bei Average over period (Durchschnitt über den Zeitraum) aus, wie viele Metrikdaten für die einzelnen Varianten gesammelt werden sollen, bevor sie verglichen werden.

Sie können wieder Add new alarm (Neuen Alarm hinzufügen) auswählen, um weitere Alarme zum Experiment hinzuzufügen.

Wählen Sie anschließend die Option Set notifications for the alarm (Benachrichtigungen für den Alarm einstellen) aus und wählen oder erstellen Sie ein Amazon-Simple-Notification-Service-Thema zum Senden von Alarmbenachrichtigungen. Weitere Informationen finden Sie unter Einrichten von Amazon-SNS-Benachrichtigungen,

10. (Optional) Um diesem Experiment Tags hinzuzufügen, wählen Sie Tags und Add new tag (Neues Tag hinzufügen) aus.

    Geben Sie für Key (Schlüssel) einen Namen für das Tag ein. Sie können einen optionalen Wert für das Tag unter Value (Wert) hinzufügen.

    (Optional) Zum Hinzufügen eines weiteren Tags wählen Sie Add new tag (Neues Tag hinzufügen) erneut aus.

    Weitere Informationen finden Sie unter AWS Ressourcen taggen.
11. Wählen Sie Create experiment (Experiment erstellen).
12. Wenn Sie dies noch nicht getan haben, bauen Sie die Feature-Varianten in Ihre Anwendung ein.
13. Wählen Sie Done (Erledigt) aus. Das Experiment beginnt erst, wenn Sie es starten.

Nachdem Sie die folgenden Schritte abgeschlossen haben, beginnt das Experiment sofort.

Ein erstelltes Experiment starten

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Evidently aus.
3. Wählen Sie den Namen des Projekts aus.
4. Wechseln Sie zur Registerkarte Experiments (Experimente).
5. Wählen Sie die Schaltfläche neben dem Namen des Experiments und dann Actions (Aktionen) und Start experiment (Experiment starten) aus.
6. (Optional) Um die bei der Erstellung vorgenommenen Versuchseinstellungen anzuzeigen oder zu ändern, wählen Sie Experiment setup (Einrichtung des Experiments) aus.
7. Wählen Sie einen Zeitpunkt, an dem das Experiment beendet werden soll.
8. Wählen Sie Start Experiment (Experiment starten) aus.

Das Experiment beginnt sofort.

Verwalten von Funktionen, Starts und Experimenten

Gehen Sie wie in diesen Abschnitten beschrieben vor, um die von Ihnen erstellten Funktionen, Starts und Experimente zu verwalten.

Themen

Siehe die aktuellen Bewertungsregeln und den Zielgruppenverkehr für ein Feature

Sie können die CloudWatch Evidenly-Konsole verwenden, um zu sehen, wie die Bewertungsregeln der Funktion den Zielgruppen-Traffic auf die aktuellen Starts, Experimente und Varianten der Funktion verteilen.

Den Zielgruppen-Datenverkehr für ein Feature anzeigen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Evidently aus.
3. Wählen Sie den Namen des Projekts aus, das das Feature enthält.
4. Wählen Sie die Registerkarte Features (Funktionen) aus.
5. Wählen Sie den Namen des Features aus.

   Auf der Registerkarte Evaluation rules (Regeln für die Bewertung) können Sie den Fluss des Zielgruppen-Datenverkehrs für Ihr Feature wie folgt sehen:

   - Zunächst werden die Überschreibungen ausgewertet. Diese geben an, dass bestimmten Benutzern immer eine bestimmte Variante bereitgestellt wird. Die Sitzungen von Benutzern,

denen Überschreibungen zugewiesen wurden, tragen nicht zu Start- oder Experimentmetriken bei.

- Als nächstes steht der verbleibende Datenverkehr für den laufenden Start zur Verfügung, falls vorhanden. Wenn derzeit ein Start läuft, zeigt die Tabelle im Abschnitt Launches (Starts) den Startnamen und den Start-Datenverkehr an, der zwischen den Feature-Varianten aufgeteilt ist. Rechts vom Abschnitt Launches (Starts) ist die Anzeige Traffic (Datenverkehr), in der Sie sehen, wie viel der verfügbaren Zielgruppe (nach Überschreibungen) diesem Start zugewiesen ist. Der Rest des Datenverkehrs, der dem Start nicht zugewiesen ist, fließt dem Experiment (falls vorhanden) und dann der Standardvariante zu.
- Als Nächstes steht der verbleibende Datenverkehr für das laufende Experiment zur Verfügung, falls vorhanden. Wenn ein Experiment im Gange ist, zeigt die Tabelle im Abschnitt Experiments den Namen und den Fortschritt des Experiments an. Rechts vom Abschnitt Experiments ist die Anzeige Traffic (Datenverkehr), auf der Sie sehen, wie viel der verfügbaren Zielgruppe (nach Überschreibungen und Starts) diesem Experiment zugewiesen ist. Der Rest des Datenverkehrs, der dem Start bzw. dem Experiment nicht zugewiesen ist, fließt der Standardvariante des Features zu.

Ändern des Datenverkehrs für den Start

Sie können die Zuweisung des Datenverkehrs für einen Start jederzeit ändern, auch während der Start läuft.

Wenn Sie sowohl einen laufenden Start als auch ein laufendes Experiment für dasselbe Feature haben, führen Änderungen am Datenverkehr des Features zu einer Änderung des Experiment-Datenverkehrs. Das liegt daran, dass dem Experiment derjenige Teil der gesamten Zielgruppemenge zu Verfügung steht, der noch nicht dem Start zugewiesen ist. Mit zunehmendem Start-Datenverkehr verkleinert sich die Zielgruppe, die für das Experiment verfügbar ist. Umgekehrt vergrößert sie sich, wenn der Start beendet oder sein Datenverkehr verringert wird.

Den zugewiesenen Datenverkehr für einen Start ändern

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Evidently aus.
3. Wählen Sie den Namen des Projekts aus, das den Start enthält.
4. Wechseln Sie zur Registerkarte Launches (Starts).
5. Wählen Sie den Namen des Starts.

Wählen Sie Modify launch traffic (Datenverkehr für den Start ändern) aus.

6. Geben Sie bei Serve (Bereitstellen) den neuen Prozentanteil ein, der den einzelnen Varianten zugewiesen werden soll. Sie können auch festlegen, dass Varianten von der Bereitstellung an Benutzer ausgeschlossen werden. Wenn Sie diese Werte ändern, sehen Sie die Auswirkungen auf den gesamten Feature-Datenverkehr direkt in der Traffic summary (Datenverkehrsübersicht).

   In der Traffic summary (Datenverkehrsübersicht) sehen Sie, wie viel von Ihrem gesamten Datenverkehr für diesen Start verfügbar ist und wie viel von diesem verfügbaren Datenverkehr diesem Start zugewiesen wird.
7. Wählen Sie Modify (Ändern) aus.

Ändern der zukünftigen Schritte eines Starts

Sie können die Konfiguration der noch nicht stattfindenden Startschritte ändern und einem Start weitere Schritte hinzufügen.

Die Schritte für einen Start ändern

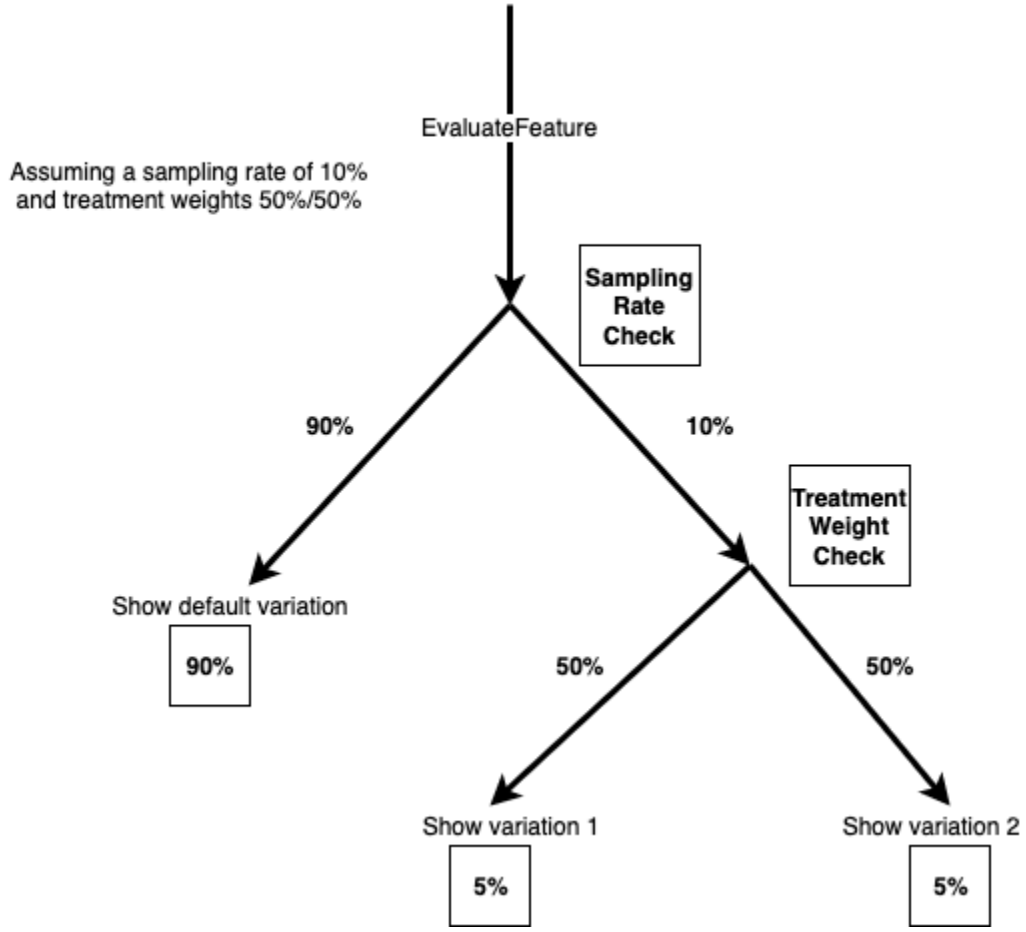
1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Evidently aus.
3. Wählen Sie den Namen des Projekts aus, das den Start enthält.
4. Wechseln Sie zur Registerkarte Launches (Starts).
5. Wählen Sie den Namen des Starts.

   Wählen Sie Modify launch traffic (Datenverkehr für den Start ändern) aus.
6. Wählen Sie Schedule launch (Start planen) aus.
7. Bei allen Schritten, die noch nicht begonnen haben, können Sie den Anteil der verfügbaren Zielgruppe ändern, der für das Experiment verwendet werden soll. Sie können auch ändern, wie deren Datenverkehr zwischen den Varianten zugewiesen wird.

   Sie können mit Add another step (Weiteren Schritt hinzufügen) dem Start weitere Schritte hinzufügen. Ein Start kann maximal fünf Schritte haben.
8. Wählen Sie Ändern aus.

Ändern des Datenverkehrs für das Experiment

Sie können die Abtastrate für ein Experiment jederzeit ändern, auch während das Experiment läuft. Sie können die Behandlungsgewichte jedoch nicht aktualisieren, nachdem ein Experiment durchgeführt wurde. Daher können Sie den gesamten Datenverkehr, der dem Experiment ausgesetzt ist, nach der Durchführung eines Experiments ändern, nicht jedoch die relative Zuordnung zu den einzelnen Behandlungen. Wenn Sie den Datenverkehr eines laufenden Experiments ändern, empfehlen wir Ihnen, die Zuweisung des Datenverkehrs nur zu erhöhen, damit es zu keiner Verzerrung kommt.



Den zugewiesenen Datenverkehr für ein Experiment ändern

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application monitoring (Anwendungsüberwachung) und Evidently aus.
3. Wählen Sie den Namen des Projekts aus, das den Start enthält.
4. Wechseln Sie zur Registerkarte Experiments (Experimente).

5. Wählen Sie den Namen des Starts.
6. Wählen Sie Modify experiment traffic (Datenverkehr für das Experiment ändern) aus.
7. Geben Sie ein, welchen prozentualen Anteil des verfügbaren Datenverkehrs diesem Experiment zugewiesen werden soll oder benutzen Sie den Schieberegler. Der verfügbare Datenverkehr ist die gesamte Zielgruppe abzüglich des Datenverkehrs, der einem aktuellen Start zugewiesen wird, falls es einen gibt. Der Datenverkehr, der dem Start bzw. dem Experiment nicht zugewiesen ist, fließt der Standardvariante zu.
8. Wählen Sie Ändern aus.

Stoppen eines Starts

Wenn Sie einen laufenden Start stoppen, können Sie ihn nicht fortsetzen oder neu starten. Außerdem wird er nicht als Regel für die Zuweisung des Datenverkehrs ausgewertet. Der Datenverkehr, der dem Start zugewiesen war, steht stattdessen dem Experiment der Funktion zur Verfügung, falls es eines gibt. Ansonsten wird der gesamte Datenverkehr nach dem Stoppen des Starts für die Standardvariante bereitgestellt.

Einen Start dauerhaft stoppen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Evidently aus.
3. Wählen Sie den Namen des Projekts aus, das den Start enthält.
4. Wechseln Sie zur Registerkarte Launch (Start).
5. Wählen Sie die Schaltfläche links neben dem Namen des Starts aus.
6. Wählen Sie Actions (Aktionen) und Cancel launch (Start abbrechen) bzw. Actions und Mark as complete (Als abgeschlossen markieren) aus.

Stoppen eines Experiments

Wenn Sie ein laufendes Experiment stoppen, können Sie es nicht fortsetzen oder neu starten. Der Teil des Datenverkehrs, der zuvor für das Experiment verwendet wurde, wird der Standardvariante bereitgestellt.

Wenn ein Experiment nicht manuell gestoppt wird und es sein Enddatum überschreitet, ändert sich der Datenverkehr nicht. Der dem Experiment zugewiesene Teil des Datenverkehrs fließt immer

noch zum Experiment. Um es zu stoppen und zu dafür zu sorgen, dass der Datenverkehr des Experiments stattdessen der Standardvariante bereitgestellt wird, markieren Sie das Experiment als abgeschlossen.

Wenn Sie ein Experiment stoppen, können Sie es abbrechen oder als abgeschlossen markieren. Wenn Sie es abbrechen, erscheint es in der Liste der Experimente als Cancelled (Abgebrochen). Wenn Sie es als abgeschlossen kennzeichnen, ist es Completed (Abgeschlossen).

Ein Experiment dauerhaft stoppen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Evidently aus.
3. Wählen Sie den Namen des Projekts aus, das das Experiment enthält.
4. Wechseln Sie zur Registerkarte Experiments (Experimente).
5. Wählen Sie die Schaltfläche links neben dem Namen des Experiments aus.
6. Wählen Sie Actions (Aktionen) und Cancel experiment (Experiment abbrechen) bzw. Actions und Mark as complete (Als abgeschlossen markieren) aus.

Hinzufügen von Programmcode zur Anwendung

Um mit CloudWatch Evidently zu arbeiten, fügen Sie Ihrer Anwendung Code hinzu, um jeder Benutzersitzung eine Variante zuzuweisen und Metriken an Evidently zu senden. Sie verwenden den `EvaluateFeature` Vorgang CloudWatch Eviently, um Benutzersitzungen Varianten zuzuweisen, und Sie verwenden den `PutProjectEvents` Vorgang, um Ereignisse an Evidently zu senden, damit diese dann Metriken für Ihre Starts oder Experimente berechnen können.

Wenn Sie Varianten oder benutzerdefinierte Metriken erstellen, bietet die CloudWatch Evidently-Konsole Beispiele für den Code, den Sie hinzufügen müssen.

Ein end-to-end Beispiel finden Sie unterTutorial: A/B-Tests mit der Evidently-Beispielanwendung.

Verwenden EvaluateFeature

Wenn bei einem Start oder Experiment Funktionsvariationen verwendet werden, verwendet die Anwendung diesen EvaluateFeatureVorgang, um jeder Benutzersitzung eine Variante zuzuweisen. Die Zuweisung einer Variante zu einem Benutzer ist ein Auswertungsereignis. Wenn Sie diesen Vorgang aufrufen, geben Sie Folgendes ein:

- Feature name (Name der Funktion) – Erforderlich. Evidently verarbeitet die Bewertung gemäß den Feature-Evaluierungsregeln des Starts bzw. des Experiments und wählt eine Variante für die Entität aus.
- entityId – Erforderlich. Repräsentiert einen eindeutigen Benutzer.
- evaluationContext – Optional. Ein JSON-Objekt, das zusätzliche Informationen zu einem Benutzer darstellt. Dieser Wert wird von Evidently verwendet, um den Benutzer bei Feature-Auswertungen mit einem Segment Ihrer Zielgruppe abzugleichen, wenn Sie Segmente erstellt haben. Weitere Informationen finden Sie unter Fokussieren Ihrer Zielgruppe mithilfe von Segmenten.

  Im folgenden Beispiel wird ein Wert vom Typ `evaluationContext` verwendet, den Sie an Evidently senden können:

```
{
    "Browser": "Chrome",
    "Location": {
        "Country": "United States",
        "Zipcode": 98007
    }
}
```

Sticky Evaluations

CloudWatch Verwendet offensichtlich „feste“ Evaluationen. Eine einzige Konfiguration von `entityId`, Feature, Feature-Konfiguration und `evaluationContext` erhält immer dieselbe Variantenzuweisung. Diese Zeitvariationszuweisung ändert sich nur dann, wenn eine Entität zu einer Überschreibung hinzugefügt wird oder der Versuchsverkehr gewählt wird.

Eine Featurekonfiguration umfasst Folgendes:

- Die Feature-Variationen
- Die Variantenkonfiguration (Prozentsätze, die jeder Variante zugewiesen sind) für ein aktuell ausgeführtes Experiment für dieses Feature, falls vorhanden.
- Die Variantenkonfiguration für einen aktuell ausgeführten Start für dieses Feature, falls vorhanden. Die Variantenkonfiguration umfasst gegebenenfalls die definierten Segmentüberschreibungen.

Wenn die Datenverkehrszuordnung für ein Experiment erhöht wird, erhalten alle `entityId`, die zuvor einer Experimentbehandlungsgruppe zugewiesen wurden, weiterhin dieselbe Behandlung.

Jede `entityId`, die zuvor der Kontrollgruppe zugewiesen war, kann gemäß der für das Experiment angegebenen Variationskonfiguration einer Experimentbehandlungsgruppe zugewiesen werden.

Wenn die Datenverkehrszuordnung eines Experiments verringert wird, kann eine `entityId` zwar von einer Behandlungsgruppe in eine Kontrollgruppe übergehen, jedoch nicht in eine andere Behandlungsgruppe.

Verwenden PutProjectEvents

Um eine benutzerdefinierte Metrik für Evidently zu codieren, verwenden Sie die PutProjectEventsOperation. Im Folgenden sehen Sie ein einfaches Beispiel für eine Nutzlast.

```
{
    "events": [
        {
            "timestamp": {{$timestamp}},
            "type": "aws.evidently.custom",
            "data": "{\"details\": {\"pageLoadTime\": 800.0}, \"userDetails\":
 {\"userId\": \"test-user\"}}"
        }
    ]
}
```

Der `entityIdKey` kann einfach ein `entityId` sein oder Sie benennen ihn um, etwa zu `userId`. Im eigentlichen Fall kann `entityId` ein Benutzername, eine Sitzungs-ID usw. sein.

```
"metricDefinition":{
    "name": "noFilter",
    "entityIdKey": "userDetails.userId", //should be consistent with jsonValue in
 events "data" fields
    "valueKey": "details.pageLoadTime"
},
```

Um sicherzustellen, dass Ereignisse mit dem richtigen Start oder Experiment verbunden sind, müssen Sie dasselbe `entityId` übergeben, wenn Sie sowohl `EvaluateFeature` als auch `PutProjectEvents` aufrufen. Stellen Sie sicher, dass Sie `PutProjectEvents` nach dem `EvaluateFeature` Anruf anrufen, da sonst Daten gelöscht werden und von CloudWatch Eviently nicht verwendet werden.

Die `PutProjectEvents`-Operation benötigt den Feature-Namen nicht als Eingabeparameter. Auf diese Weise können Sie ein einzelnes Ereignis in mehreren Experimenten verwenden. Angenommen

Sie rufen `EvaluateFeature` auf, wobei `entityId` auf `userDetails.userId` gesetzt ist. Wenn Sie zwei oder mehr Experimente ausführen, können Sie ein einzelnes Ereignis aus der Sitzung dieses Benutzers Metriken für jedes dieser Experimente ausgeben lassen. Rufen Sie hierfür `PutProjectEvents` einmal für jedes Experiment mit demselben `entityId` auf.

Timing

Nachdem Ihre Bewerbung `EvaluateFeature` aufruft, gibt es einen Zeitraum von einer Stunde, in dem Metrikereignisse von `PutProjectEvents` auf der Grundlage dieser Bewertung zugeschrieben werden. Wenn nach einem Zeitraum von einer Stunde weitere Ereignisse auftreten, werden diese nicht zugeordnet.

Wenn jedoch der gleiche `entityId` für einen neuen `EvaluateFeature`-Aufruf während des Zeitraums von einer Stunde verwendet wird, wird jetzt stattdessen das spätere `EvaluateFeature`-Ergebnis verwendet, und der einstündige Timer wird neu gestartet. Dies kann nur unter bestimmten Umständen geschehen, z. B. wenn der Testdatenverkehr zwischen den beiden Zuweisungen eingewählt wird, wie im vorherigen Abschnitt Sticky Evaluations erläutert.

Ein end-to-end Beispiel finden Sie unterTutorial: A/B-Tests mit der Evidently-Beispielanwendung.

Projekt-Datenspeicherung

Evidently sammelt zwei Arten von Ereignissen:

- Auswertungsereignisse – Sie hängen damit zusammen, welche Variante eines Features einer Benutzersitzung zugewiesen ist. Evidently verwendet diese Ereignisse, um Metriken und andere Experiment- und Startdaten zu erstellen, die Sie in der Evidently-Konsole sehen können.

  Sie können diese Evaluierungsereignisse auch in Amazon CloudWatch Logs oder Amazon S3 speichern.
- Benutzerdefinierte Ereignisse – Sie werden verwendet, um Metriken aus Benutzeraktionen wie Klicks und Kaufabwicklungen zu generieren. Evidently bietet keine Methode zum Speichern von benutzerdefinierten Ereignissen. Wenn Sie sie speichern möchten, müssen Sie Ihren Anwendungscode ändern, um sie an einen Speicher außerhalb von Evidently zu senden.

Format der Evaluierungsereignisprotokolle

Wenn Sie sich dafür entscheiden, Evaluierungsereignisse in CloudWatch Logs oder Amazon S3 zu speichern, wird jedes Evaluierungsereignis als Protokollereignis mit dem folgenden Format gespeichert:

```
{
    "event_timestamp": 1642624900215,
    "event_type": "evaluation",
    "version": "1.0.0",
    "project_arn": "arn:aws:evidently:us-east-1:123456789012:project/petfood",
    "feature": "petfood-upsell-text",
    "variation": "Variation1",
    "entity_id": "7",
    "entity_attributes": {},
    "evaluation_type": "EXPERIMENT_RULE_MATCH",
    "treatment": "Variation1",
    "experiment": "petfood-experiment-2"
}
```

Hier sind weitere Details zum vorhergehenden Evaluierungsereignisformat:

- Der Zeitstempel ist in UNIX-Zeit mit Millisekunden
- Die Variante ist der Name der Variante des Features, das dieser Benutzersitzung zugewiesen wurde.
- Die Entitäts-ID ist eine Zeichenfolge.
- Entitätsattribute sind ein Hash beliebiger Werte, die vom Client gesendet werden. Zum Beispiel, wenn`entityId` blau oder grün zugeordnet ist, dann können Sie optional UserIDs, Sitzungsdaten oder was auch immer Sie möchten aus Korrelation- und Data Warehouse-Perspektive senden.

IAM-Richtlinie und Verschlüsselung für Evaluierungsereignisspeicher in Simple Storage Service (Amazon S3)

Wenn Sie sich für die Verwendung von Amazon S3 entscheiden, um Evaluierungsereignisse zu speichern, müssen Sie eine IAM-Richtlinie wie die folgende hinzufügen, damit Evidently Protokolle im Amazon S3-Bucket veröffentlichen kann. Dies liegt daran, dass Amazon S3 Buckets und die darin enthaltenen Objekte privat sind und standardmäßig keinen Zugriff auf andere Dienste zulassen.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {"Service": "delivery.logs.amazonaws.com"},
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::bucket_name/optional_folder/AWSLogs/account_id/
*",
            "Condition": {"StringEquals": {"s3:x-amz-acl": "bucket-owner-full-
control"}}
        },
        {
            "Sid": "AWSLogDeliveryCheck",
            "Effect": "Allow",
            "Principal": {"Service": "delivery.logs.amazonaws.com"},
            "Action": ["s3:GetBucketAcl", "s3:ListBucket"],
            "Resource": "arn:aws:s3:::bucket_name"
        }
    ]
}
```

Wenn Sie Evidently-Daten in Amazon S3 speichern, können Sie sie auch mit serverseitiger Verschlüsselung mit AWS Key Management Service -Schlüsseln (SSE-KMS) sichern. Weitere Informationen finden Sie unter Schützen von Daten mithilfe serverseitiger Verschlüsselung.

Wenn Sie einen vom Kunden verwalteten Schlüssel von verwenden AWS KMS, müssen Sie der IAM-Richtlinie für Ihren Schlüssel Folgendes hinzufügen. Dies ermöglicht es Evidently, in den Bucket zu schreiben.

```
{
    "Sid": "AllowEvidentlyToUseCustomerManagedKey",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "delivery.logs.amazonaws.com"
        ]
    },
   "Action": [
       "kms:Encrypt",
       "kms:Decrypt",
       "kms:ReEncrypt*",
       "kms:GenerateDataKey*",
```

```
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

Ergebnisberechnung von Evidently

Sie können Amazon CloudWatch Evidently A/B-Tests als Tool für datengestützte Entscheidungsfindung verwenden. In einem A/B-Test werden die Benutzer nach dem Zufallsprinzip entweder der Kontrollgruppe (auch Standardvariante genannt) oder einer der Behandlungsgruppen (auch getestete Variante genannt) zugewiesen. Beispielsweise könnten Benutzer in der Kontrollgruppe die Website, den Dienst oder die Anwendung auf die gleiche Weise wie vor Beginn des Experiments erleben. In der Zwischenzeit können Benutzer in der Behandlungsgruppe die Veränderung erleben.

CloudWatch Unterstützt offenbar bis zu fünf verschiedene Varianten in einem Experiment. Evidently weist diesen Varianten zufällig Traffic zu. Auf diese Weise können Sie Geschäftsmetriken (z. B. Umsatz) und Leistungsmetriken (z. B. Latenz) für jede Gruppe nachverfolgen. Evidently geht folgendermaßen vor:

- Vergleich der Behandlung mit der Kontrolle. (Vergleicht beispielsweise, ob der Umsatz mit einem neuen Checkout-Prozess steigt oder sinkt.)
- Gibt an, ob der beobachtete Unterschied zwischen der Behandlung und der Kontrolle signifikant ist. Dafür bietet Evidently zwei Lösungsansätze: frequentistische Signifikanzniveaus und Bayessche Wahrscheinlichkeiten.

Welche Gründe sprechen für frequentistische und Bayessche Ansätze?

Stellen Sie sich einen Fall vor, in dem die Behandlung im Vergleich zur Kontrolle keine Wirkung hat, oder einen Fall, in dem die Behandlung mit der Kontrolle identisch ist (ein A/A-Test). Sie würden immer noch einen kleinen Unterschied zwischen der Behandlung und der Kontrolle in den Daten feststellen. Dies liegt daran, dass die Gruppe der Testteilnehmer aus einer endlichen Stichprobe von Benutzern besteht, die einen kleinen Prozentsatz aller Benutzer der Website, des Service oder der Anwendung ausmachen. Frequentistische Signifikanzniveaus und Bayessche Wahrscheinlichkeiten geben Aufschluss darüber, ob der beobachtete Unterschied signifikant oder zufällig ist.

Evidently berücksichtigt die folgenden Kriterien bei der Feststellung, ob der beobachtete Unterschied signifikant ist:

- Wie groß der Unterschied ist
- Wie viele Proben Teil des Tests sind
- Wie die Daten verteilt werden

Frequentistische Analyse in Evidently

Evidently verwendet sequentielle Tests, wodurch die üblichen Peeking-Probleme vermieden werden, ein häufiger Fallstrick für die frequentistische Statistik. Beim Peeking werden die Ergebnisse eines laufenden A/B-Tests überprüft, um ihn zu stoppen und auf der Grundlage der beobachteten Ergebnisse eine Entscheidung zu treffen. Weitere Informationen zu sequentiellen Tests finden Sie unter Time-uniform, nonparametric, nonasymptotic confidence sequences (Zeiteinheitliche, nicht parametrische, nicht asymptotische Konfidenzsequenzen) von Howard et al. (Ann. Statist. 49 (2) 1055–1080, 2021).

Weil die Ergebnisse von Evidently jederzeit gültig sind (jederzeit gültige Ergebnisse), können Sie sich die Ergebnisse während des Experiments ansehen („Peeking“) und trotzdem fundierte Schlussfolgerungen ziehen. Das kann einen Teil der Experimentierkosten reduzieren, da Sie ein Experiment vor dem geplanten Zeitpunkt beenden können, wenn die Ergebnisse bereits signifikant sind.

Evidently erzeugt jederzeit gültige Signifikanzniveaus und jederzeit gültige 95-%-Konfidenzintervalle der Differenz zwischen der getesteten Variante und der Standardvariante in der Zielmetrik. Die Spalte Result (Ergebnis) in den Versuchsergebnissen gibt die Leistung der getesteten Variante an. Folgende Optionen sind möglich:

- Inconclusive (Nicht eindeutig) – Das Signifikanzniveau liegt unter 95 %
- Better (Besser) – Das Signifikanzniveau liegt bei 95 % oder höher und eine der folgenden Aussagen trifft zu:
  - Die Untergrenze des 95-%-Konfidenzintervalls ist höher als Null und die Metrik sollte steigen
  - Die Obergrenze des 95-%-Konfidenzintervalls ist niedriger als Null und die Metrik sollte abnehmen
- Worse (Schlechter) – Das Signifikanzniveau liegt bei 95 % oder höher und eine der folgenden Aussagen trifft zu:
  - Die Obergrenze des 95-%-Konfidenzintervalls ist höher als Null und die Metrik sollte steigen
  - Die Untergrenze des 95-%-Konfidenzintervalls ist niedriger als Null und die Metrik sollte abnehmen

- Best (Am besten) – Das Experiment hat zusätzlich zur Standardvariante zwei oder mehr getestete Varianten, und die folgenden Bedingungen sind erfüllt:
  - Die Variante qualifiziert sich für die Bezeichnung Better (Besser)
  - Eine der folgenden Bedingungen trifft zu:
    - Die Untergrenze des 95-%-Konfidenzintervalls ist höher als die Obergrenze der 95-%-Konfidenzintervalle aller anderen Variationen, und die Metrik sollte steigen
    - Die Obergrenze des 95-%-Konfidenzintervalls ist niedriger als die Obergrenze der 95-%-Konfidenzintervalle aller anderen Variationen, und die Metrik sollte abnehmen

Bayessche Analyse in Evidently

Mit der Bayesschen Analyse können Sie die Wahrscheinlichkeit berechnen, dass der Mittelwert in der getesteten Variante größer oder kleiner als der Mittelwert der Standardvariante ist. Evidently führt eine Bayessche Inferenz für den Mittelwert der Zielmetrik durch, indem konjugierte Priore verwendet werden. Mit konjugierten Prioren kann Evidently effizienter auf die spätere Verteilung schließen, die für die Bayessche Analyse erforderlich ist.

Evidently wartet bis zum Enddatum des Experiments, um die Ergebnisse der Bayesschen Analyse zu berechnen. Auf der Ergebnisseite wird Folgendes angezeigt:

- probability of increase (Wahrscheinlichkeit einer Erhöhung) – Die Wahrscheinlichkeit, dass der Mittelwert der Metrik in der getesteten Variante mindestens 3 % größer als der Mittelwert der Standardvariante ist
- probability of decrease (Wahrscheinlichkeit eines Rückgangs) – Die Wahrscheinlichkeit, dass der Mittelwert der Metrik in der getesteten Variante mindestens 3 % kleiner als der Mittelwert der Standardvariante ist
- probability of no change (Wahrscheinlichkeit keiner Änderung) – Die Wahrscheinlichkeit, dass der Mittelwert der Metrik in der getesteten Variante innerhalb von ± 3 % des Mittelwerts der Standardvariante liegt

Die Spalte Result (Ergebnis) gibt die Leistung der Variante an. Folgende Optionen sind möglich:

- Better (Besser) – Die Wahrscheinlichkeit eines Anstiegs beträgt mindestens 90 % und die Metrik sollte steigen, oder die Wahrscheinlichkeit eines Rückgangs beträgt mindestens 90 % und die Metrik sollte abnehmen

- Worse (Schlechter) – Die Wahrscheinlichkeit eines Rückgangs beträgt mindestens 90 % und die Metrik sollte steigen, oder die Wahrscheinlichkeit eines Anstiegs beträgt mindestens 90 % und die Metrik sollte abnehmen

Anzeigen von Launch-Ergebnissen im Dashboard

Sie können die Fortschritts- und Metrikergebnisse eines Experiments sehen, während es läuft und nachdem es abgeschlossen ist.

Den Fortschritt und die Ergebnisse eines Starts ansehen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Evidently aus.
3. Wählen Sie den Namen des Projekts aus, das den Start enthält.
4. Wechseln Sie zur Registerkarte Launch (Start).
5. Wählen Sie den Namen des Starts.
6. Um die Startschritte und die Datenverkehrs-Zuweisungen für die einzelnen Schritte zu sehen, wählen Sie die Registerkarte Launch (Start).
7. Um die Anzahl der Benutzersitzungen anzuzeigen, die den einzelnen Varianten im Laufe der Zeit zugewiesen sind, sowie um die Leistungsmetriken für die einzelnen Varianten des Starts anzuzeigen, wählen Sie die Registerkarte Monitoring (Überwachung) aus.

   In dieser Ansicht sehen Sie auch, ob Startalarme während des Starts in den Status `ALARM` gewechselt sind.
8. Um die Varianten, Metriken, Alarme und Tags für diesen Start zu sehen, wählen Sie die Registerkarte Configuration (Konfiguration) aus.

Anzeigen von Versuchsergebnissen im Dashboard

Sie können die statistischen Ergebnisse eines Experiments sehen, während es läuft und nachdem es abgeschlossen ist. Die Ergebnisse eines Experiments sind bis zu 63 Tage nach dem Start des Experiments verfügbar. Danach sind sie aufgrund von Richtlinien zur CloudWatch Datenspeicherung nicht mehr verfügbar.

Es werden keine statistischen Ergebnisse angezeigt, bis jede Variante mindestens 100 Ereignisse aufweist.

Evidently führt am Ende des Experiments eine zusätzliche Offline-p-Wert-Analyse durch. Offline-p-Wert-Analysen können in einigen Fällen, in denen die während des Experiments verwendeten zeitunabhängigen p-Werte keine statistische Signifikanz finden, statistische Signifikanz erkennen.

Weitere Informationen darüber, wie CloudWatch Evidently die Versuchsergebnisse berechnet, finden Sie unter. Ergebnisberechnung von Evidently

Die Ergebnisse eines Experiments ansehen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Evidently aus.
3. Wählen Sie den Namen des Projekts aus, das das Experiment enthält.
4. Wechseln Sie zur Registerkarte Experiments (Experimente).
5. Wählen Sie den Namen des Experiments und dann die Registerkarte Results (Ergebnisse).
6. Bei Variation performance (Leistung der Variante) gibt es ein Steuerelement, mit dem Sie auswählen können, welche Experimentstatistiken angezeigt werden sollen. Wenn Sie mehr als eine Statistik auswählen, zeigt Evidently ein Diagramm und eine Tabelle für eine jede Statistik an.

   Die einzelnen Diagramme und Tabellen zeigen die bisherigen Ergebnisse des Experiments an.

   Die Diagramme können die folgenden Ergebnisse anzeigen. Sie können mit dem Steuerelement rechts neben dem Diagramm bestimmen, welches der folgenden Elemente angezeigt wird:

   - Die Anzahl der für jede Variante aufgezeichneten Benutzersitzungsereignisse.
   - Der Durchschnittswert der Metrik der einzelnen Varianten, die oben im Diagramm ausgewählt wird.
   - Das statistische Gewicht der Experimente. Hier wird die Differenz für die oben im Diagramm ausgewählte Metrik mit der Standardvariante und allen anderen Varianten verglichen.
   - Das obere und untere Vertrauensniveau von 95 % begrenzt die Differenz der ausgewählten Metrik zwischen jeder Variante und der Standardvariante.

   Die Tabelle hat eine Zeile pro Variante. Für jede Variante, die nicht die Standardvariante ist, zeigt Evidently an, ob es genügend Daten erhalten hat und die Ergebnisse dadurch statistisch aussagekräftig sind. Sie sehen auch, ob die Verbesserung des statistischen Werts durch die Veränderung ein Vertrauensniveau von 95 % erreicht hat.

Zu guter Letzt sehen Sie in der Spalte Result (Ergebnis) eine Empfehlung, welche Variante basierend auf dieser Statistik am besten abschneidet bzw. ob es sich nicht eindeutig sagen lässt.

Wie sammelt und speichert CloudWatch Eviently Daten

Amazon sammelt und speichert CloudWatch offensichtlich Daten zu Projektkonfigurationen, sodass Kunden Experimente und Produkteinführungen durchführen können. Die Daten umfassen Folgendes:

- Metadaten über Projekte, Funktionen, Starts und Experimente
- Metrikereignisse
- Auswertungsdaten

Ressourcen-Metadaten werden in Amazon DynamoDB gespeichert. Die Daten werden im Ruhezustand standardmäßig verschlüsselt, und zwar mit AWS-eigene Schlüssel. Bei diesen Schlüsseln handelt es sich um eine Sammlung von AWS KMS Schlüsseln, die ein AWS-Service Unternehmen besitzt und verwaltet, sodass sie in mehreren Fällen verwendet AWS-Konten werden können. Kunden können die Verwendung dieser Schlüssel nicht einsehen, verwalten oder prüfen. Kunden müssen auch keine Maßnahmen ergreifen oder die Programme anpassen, die die Schlüssel zur Datenverschlüsselung schützen.

Weitere Informationen finden Sie AWS-eigene Schlüsselim AWS Key Management Service Entwicklerhandbuch.

Evidently-Metrikereignisse und -Evaluierungsereignisse werden direkt an kundeneigene Speicherorte gesendet.

Daten werden während der Übertragung automatisch mit HTTPS verschlüsselt. Diese Daten werden an kundeneigene Speicherorte gesendet.

Sie können sich auch dafür entscheiden, Evaluierungsereignisse in Amazon Simple Storage Service oder Amazon CloudWatch Logs zu speichern. Weitere Informationen darüber, wie Sie Ihre Daten in diesen Services schützen können, finden Sie unter Aktivieren der Amazon S3 S3-Standard-Bucket-Verschlüsselung und Verschlüsseln von Protokolldaten in CloudWatch Logs using AWS KMS.

Abrufen von Daten

Sie können Ihre Daten mithilfe von CloudWatch Evidenly-APIs abrufen. Um Projektdaten abzurufen, verwenden Sie GetProjectoder ListProjects.

Verwenden Sie GetFeatureoder, um Objektdaten abzurufen ListFeatures.

Verwenden Sie GetLaunchoder, um Startdaten abzurufen ListLaunches.

Verwenden GetExperimentSie, oder, um Versuchsdaten abzurufen GetExperimentResults. ListExperiments

Ändern und Löschen von Daten

Sie können Ihre Daten mithilfe von CloudWatch Evidence-APIs ändern und löschen. Verwenden Sie für Projektdaten UpdateProjectoder DeleteProject.

Verwenden Sie für Feature-Daten UpdateFeatureoder DeleteFeature.

Verwenden Sie für Startdaten UpdateLaunchoder DeleteLaunch.

Verwenden Sie für Versuchsdaten UpdateExperimentoder DeleteExperiment.

Verwendung von serviceverknüpften Rollen für Evidently

CloudWatch Verwendet offensichtlich AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Evidently verknüpft ist. Dienstbezogene Rollen sind von Evidently vordefiniert und beinhalten alle Berechtigungen, die der Dienst benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen.

Eine serviceverknüpfte Rolle vereinfacht das Einrichten von Evidently, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Evidently definiert die Berechtigungen seiner serviceverknüpften Rollen. Sofern keine andere Konfiguration festgelegt wurde, kann nur Evidently die Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauensrichtlinie und die Berechtigungsrichtlinie, und diese Berechtigungsrichtlinie kann keiner anderen juristischen Stelle von IAM zugeordnet werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem die zugehörigen Ressourcen gelöscht wurden. Das schützt Ihre Evidently-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rollen angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Berechtigungen von serviceverknüpften Rollen für Evidently

Verwendet offensichtlich die serviceverknüpfte Rolle mit dem Namen AWSServiceRoleForCloudWatchEvidently— Ermöglicht es CloudWatch Evidently, zugehörige AWS Ressourcen im Namen des Kunden zu verwalten.

Die AWSServiceRoleForCloudWatchEvidently dienstbezogene Rolle vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `CloudWatch Evidently`

Die genannte Rollenberechtigungsrichtlinie AmazonCloudWatchEvidentlyServiceRolePolicy ermöglicht es Evidenly, die folgenden Aktionen an den angegebenen Ressourcen durchzuführen:

- Aktionen: `appconfig:StartDeployment`, `appconfig:StopDeployment`, `appconfig:ListDeployments` und `appconfig:TagResource` bei Thick-Clients von Evidently.

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für Evidently

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie beginnen, einen Eviently Thick-Client in der AWS Management Console, der oder der AWS API zu verwenden AWS CLI, erstellt Evidently die dienstbezogene Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie beginnen, einen Evidently-Thick-Client zu verwenden, erstellt Evidently die serviceverknüpfte Rolle erneut für Sie.

Bearbeiten einer serviceverknüpften Rolle für Evidently

Erlaubt Ihnen offensichtlich nicht, die dienstbezogene Rolle zu bearbeiten. AWSServiceRoleForCloudWatchEvidently Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für Evidently

Wenn Sie ein Feature oder einen Dienst, die bzw. der eine serviceverknüpften Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpften Rolle zunächst bereinigen, bevor Sie sie manuell löschen können. Sie müssen alle Evidently-Projekte löschen, die Thick-Clients verwenden.

> **Note**
>
> Verwendet der Evidently-Service die Rolle beim Versuch, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um offensichtlich Ressourcen zu löschen, die verwendet werden von AWSServiceRoleForCloudWatchEvidently

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application monitoring (Anwendungsüberwachung) und Evidently aus.
3. Aktivieren Sie in der Liste der Projekte das Kontrollkästchen neben den Projekten, die Thick-Clients verwendet haben.
4. Wählen Sie Project actions (Projektaktionen), Delete project (Projekt löschen) aus.

So löschen Sie die serviceverknüpfte Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die AWSServiceRoleForCloudWatchEvidently serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für serviceverknüpfte Evidently-Rollen

Evidently unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter AWS -Regionen und -Endpunkte.

CloudWatch Offensichtlich Quoten

CloudWatch Hat offenbar die folgenden Quoten.

Ressource	Standardkontingent
Projekte	50 pro Region und Konto Sie können eine Kontingenterhöhung beantragen.
Segmente	500 pro Region und Konto Sie können eine Kontingenterhöhung beantragen.
Kontingente pro Projekt	• 100 Funktionen insgesamt • 500 Starts insgesamt • 50 laufende Starts • 500 Experimente insgesamt • 50 laufende Experimente Sie können für alle diese Kontingente eine Kontingen tserhöhung beantragen.
API-Quoten (alle Quoten gelten pro Region)	• PutProjectEvents: 1000 Transaktionen pro Sekunde (TPS) in den USA Ost (Nord-Virginia), den USA West (Oregon) und Europa (Irland). 200 TPS in allen anderen Regionen. • EvaluateFeature: 1000 TPS in den USA Ost (Nord-Vir ginia), den USA West (Oregon) und Europa (Irland). 200 TPS in allen anderen Regionen. • BatchEvaluateFeature: 50 TPS • APIs zum Erstellen, Lesen, Aktualisieren, Löschen (Create, Read, Update, Delete, CRUD): 10 TPS kombiniert über alle CRUD-APIs Sie können für alle diese Kontingente eine Kontingen tserhöhung beantragen.

Tutorial: A/B-Tests mit der Evidently-Beispielanwendung

Dieser Abschnitt enthält ein Tutorial zur Verwendung von Amazon CloudWatch Evidently für A/B-Tests. Dieses Tutorial ist die Evidently-Beispielanwendung, die eine einfache React-Anwendung ist. Die Beispiel-App wird so konfiguriert, dass sie entweder ein `showDiscount`-Feature anzeigt oder nicht. Wenn das Feature einem Benutzer angezeigt wird, wird der auf der Einkaufs-Website angeführte Preis mit 20 % Rabatt angezeigt.

Zusätzlich zur Anzeige des Rabatts für einige Benutzer und nicht für andere, werden Sie in diesem Tutorial Evidently einrichten, um Ladezeit-Metriken aus beiden Varianten zu sammeln.

> **Warning**
>
> Für dieses Szenario sind IAM-Benutzer mit programmatischem Zugriff und langfristigen Anmeldeinformationen erforderlich, was ein Sicherheitsrisiko darstellt. Um dieses Risiko zu minimieren, empfehlen wir, diesen Benutzern nur die Berechtigungen zu gewähren, die sie für die Ausführung der Aufgabe benötigen, und diese Benutzer zu entfernen, wenn sie nicht mehr benötigt werden. Die Zugriffsschlüssel können bei Bedarf aktualisiert werden. Weitere Informationen finden Sie unter Aktualisieren von Zugriffsschlüsseln im IAM-Benutzerhandbuch.

Schritt 1: Herunterladen der Beispielanwendung

Laden Sie zunächst die Evidenty-Beispielanwendung herunter.

So laden Sie die Beispielanwendung herunter

1. Laden Sie die Baiespielanwendung aus dem folgenden Amazon-S3-Bucket herunter:

   ```
   https://evidently-sample-application.s3.us-west-2.amazonaws.com/evidently-sample-shopping-app.zip
   ```

2. Entpacken Sie das Paket.

Schritt 2: Hinzufügen des Evidenty-Endpunkts und Einrichten der Anmeldeinformationen

Fügen Sie als Nächstes die Region und den Endpunkt für Evidently zur `config.js`-Datei im `src`-Verzeichnis im Beispiel-App-Paket hinzu, wie im folgenden Beispiel:

```
evidently: {
    REGION: "us-west-2",
    ENDPOINT: "https://evidently.us-west-2.amazonaws.com (https://evidently.us-
west-2.amazonaws.com/)",
},
```

Sie müssen außerdem sicherstellen, dass die Anwendung über die Berechtigung verfügt, CloudWatch Evidently aufzurufen.

So erteilen Sie der Beispiel-App Berechtigungen zum Aufrufen von Evidently

1. Verbinde dich mit deinem AWS Konto.
2. Erstellen Sie einen IAM-Benutzer und hängen Sie die AmazonCloudWatchEvidentlyFullAccessRichtlinie an diesen Benutzer an.
3. Notieren Sie sich die Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel des IAM-Benutzers, da Sie sie im nächsten Schritt benötigen.
4. Geben Sie in dieselbe `config.js`-Datei, die Sie zuvor in diesem Abschnitt geändert haben, die Werte der Zugriffsschlüssel-ID und des geheimen Zugriffsschlüssels ein, wie im folgenden Beispiel:

   ```
   credential: {
      accessKeyId: "Access key ID",
      secretAccessKey: "Secret key"
   }
   ```

   > **Important**
   >
   > Wir verwenden diesen Schritt, um die Beispiel-App so einfach wie möglich zu gestalten, damit Sie es ausprobieren können. Wir empfehlen jedoch nicht, dass Sie Ihre IAM-Benutzeranmeldeinformationen in Ihre tatsächliche Produktionsanwendung einfügen. Stattdessen empfehlen wir die Verwendung von Amazon Cognito zur Authentifizierung.

Weitere Informationen finden Sie unter Integration von Amazon Cognito mit Web- und mobilen Apps.

Schritt 3: Einrichten von Programmcode für die Feature-Auswertung

Wenn Sie CloudWatch Evidently verwenden, um eine Funktion zu bewerten, müssen Sie den EvaluateFeatureVorgang verwenden, um für jede Benutzersitzung nach dem Zufallsprinzip eine Funktionsvariante auszuwählen. Dieser Vorgang weist jeder Variante des Features Benutzersitzungen entsprechend den Prozentsätzen zu, die Sie im Experiment angegeben haben.

Den Programmcode für die Feature-Auswertung bei der Demo-App einrichten

1. Fügen Sie den Client Builder in der `src/App.jsx`-Datei hinzu, damit die Beispiel-App Evidently aufrufen kann.

```
import Evidently from 'aws-sdk/clients/evidently';
import config from './config';

const defaultClientBuilder = (
    endpoint,
    region,
) => {
    const credentials = {
        accessKeyId: config.credential.accessKeyId,
        secretAccessKey: config.credential.secretAccessKey
    }
    return new Evidently({
        endpoint,
        region,
        credentials,
    });
};
```

2. Fügen Sie dem `const` App-Codeabschnitt Folgendes hinzu, um den Client zu initiieren.

```
  if (client == null) {
      client = defaultClientBuilder(
        config.evidently.ENDPOINT,
        config.evidently.REGION,
      );
```

3. Konstruieren Sie `evaluateFeatureRequest`, indem Sie den folgenden Code hinzufügen. Dieser Code füllt den Projektnamen und den Feature-Namen, die wir später in diesem Tutorial empfehlen, vorab aus. Sie können sie durch Ihre eigenen Projekt- und Feature-Namen ersetzen, solange Sie diese Projekt- und Feature-Namen auch in der Evidenty-Konsole festlegen.

```
const evaluateFeatureRequest = {
              entityId: id,
              // Input Your feature name
              feature: 'showDiscount',
              // Input Your project name'
              project: 'EvidentlySampleApp',
            };
```

4. Fügen Sie den Code hinzu, der Evidently zur Feature-Auswertung aufrufen soll. Wenn die Anfrage gesendet wird, weist Evidently die Benutzersitzung nach dem Zufallsprinzip zu, um das `showDiscount`-Feature entweder zu sehen oder nicht.

```
client.evaluateFeature(evaluateFeatureRequest).promise().then(res => {
    if(res.value?.boolValue !== undefined) {
        setShowDiscount(res.value.boolValue);
    }
    getPageLoadTime()
 })
```

Schritt 4: Konfigurieren von Programmcode für die Experimentmetriken

Verwenden Sie für die benutzerdefinierte Metrik die Evidently-API `PutProjectEvents` zum Senden von Metrikergebnissen an Evidently. Die folgenden Beispiele verdeutlichen, wie Sie die benutzerdefinierte Metrik einrichten und Versuchsdaten an Evidently senden.

Fügen Sie die folgende Funktion hinzu, um die Ladezeit der Seite zu berechnen. Verwenden Sie `PutProjectEvents`, um die Metrikwerte an Evidently zu senden. Fügen Sie die folgende Funktion in `Home.tsx` hinzu und rufen Sie diese Funktion innerhalb der `EvaluateFeature`-API auf:

```
const getPageLoadTime = () => {
    const timeSpent = (new Date().getTime() - startTime.getTime()) * 1.000001;
    const pageLoadTimeData = `{
      "details": {
        "pageLoadTime": ${timeSpent}
      },
```

```
      "UserDetails": { "userId": "${id}", "sessionId": "${id}"}
    }`;
    const putProjectEventsRequest = {
      project: 'EvidentlySampleApp',
      events: [
        {
          timestamp: new Date(),
          type: 'aws.evidently.custom',
          data: JSON.parse(pageLoadTimeData)
        },
      ],
    };
    client.putProjectEvents(putProjectEventsRequest).promise();
  }
```

So sollte die App.js-Datei nach all den Bearbeitungen aussehen, die Sie seit dem Download vorgenommen haben.

```
import React, { useEffect, useState } from "react";
import { BrowserRouter as Router, Switch } from "react-router-dom";
import AuthProvider from "contexts/auth";
import CommonProvider from "contexts/common";
import ProductsProvider from "contexts/products";
import CartProvider from "contexts/cart";
import CheckoutProvider from "contexts/checkout";
import RouteWrapper from "layouts/RouteWrapper";
import AuthLayout from "layouts/AuthLayout";
import CommonLayout from "layouts/CommonLayout";
import AuthPage from "pages/auth";
import HomePage from "pages/home";
import CheckoutPage from "pages/checkout";
import "assets/scss/style.scss";
import { Spinner } from 'react-bootstrap';

import Evidently from 'aws-sdk/clients/evidently';
import config from './config';

const defaultClientBuilder = (
  endpoint,
  region,
) => {
  const credentials = {
    accessKeyId: config.credential.accessKeyId,
```

```
    secretAccessKey: config.credential.secretAccessKey
  }
  return new Evidently({
    endpoint,
    region,
    credentials,
  });
};

const App = () => {
  const [isLoading, setIsLoading] = useState(true);
  const [startTime, setStartTime] = useState(new Date());
  const [showDiscount, setShowDiscount] = useState(false);
  let client = null;
  let id = null;

  useEffect(() => {
    id = new Date().getTime().toString();
    setStartTime(new Date());
    if (client == null) {
      client = defaultClientBuilder(
        config.evidently.ENDPOINT,
        config.evidently.REGION,
      );
    }
    const evaluateFeatureRequest = {
      entityId: id,
      // Input Your feature name
      feature: 'showDiscount',
      // Input Your project name'
      project: 'EvidentlySampleApp',
    };

    // Launch
    client.evaluateFeature(evaluateFeatureRequest).promise().then(res => {
      if(res.value?.boolValue !== undefined) {
        setShowDiscount(res.value.boolValue);
      }
    });

    // Experiment
    client.evaluateFeature(evaluateFeatureRequest).promise().then(res => {
      if(res.value?.boolValue !== undefined) {
        setShowDiscount(res.value.boolValue);
```

```
      }
      getPageLoadTime()
    })

    setIsLoading(false);
  },[]);

  const getPageLoadTime = () => {
    const timeSpent = (new Date().getTime() - startTime.getTime()) * 1.000001;
    const pageLoadTimeData = `{
      "details": {
        "pageLoadTime": ${timeSpent}
      },
      "UserDetails": { "userId": "${id}", "sessionId": "${id}"}
    }`;
    const putProjectEventsRequest = {
      project: 'EvidentlySampleApp',
      events: [
        {
          timestamp: new Date(),
          type: 'aws.evidently.custom',
          data: JSON.parse(pageLoadTimeData)
        },
      ],
    };
    client.putProjectEvents(putProjectEventsRequest).promise();
  }
  return (
    !isLoading? (
    <AuthProvider>
      <CommonProvider>
        <ProductsProvider>
          <CartProvider>
            <CheckoutProvider>
              <Router>
                <Switch>
                  <RouteWrapper
                    path="/"
                    exact
                    component={() => <HomePage showDiscount={showDiscount}/>}
                    layout={CommonLayout}
                  />
                  <RouteWrapper
                    path="/checkout"
```

```
                    component={CheckoutPage}
                    layout={CommonLayout}
                  />
                  <RouteWrapper
                    path="/auth"
                    component={AuthPage}
                    layout={AuthLayout}
                  />
                </Switch>
              </Router>
            </CheckoutProvider>
          </CartProvider>
        </ProductsProvider>
      </CommonProvider>
    </AuthProvider> ) : (
      <Spinner animation="border" />
    )
  );
};

export default App;
```

Jedes Mal, wenn ein Benutzer die Beispiel-App besucht, wird eine benutzerdefinierte Metrik zur Analyse an Evidently gesendet. Evidently analysiert eine jede Metrik und zeigt Ergebnisse in Echtzeit im Evidently-Dashboard an. Das folgende Beispiel zeigt eine Metrik-Nutzlast.

```
[ {"timestamp": 1637368646.468, "type": "aws.evidently.custom", "data": "{\"details
\":{\"pageLoadTime\":2058.002058},\"userDetails\":{\"userId\":\"1637368644430\",
\"sessionId\":\"1637368644430\"}}" } ]
```

Schritt 5: Erstellen von Projekt, Feature und Experiment

Als Nächstes erstellen Sie das Projekt, die Funktion und das Experiment in der CloudWatch Evidenly-Konsole.

Projekt, Feature und Experiment für dieses Tutorial erstellen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Evidently aus.
3. Wählen Sie Create project (Projekt erstellen) aus und füllen Sie die Felder aus. Sie müssen **`EvidentlySampleApp`** für den Projektnamen verwenden, damit das Beispiel ordnungsgemäß

funktioniert. Wählen Sie für Evaluation event storage (Speicherung von Auswertungsereignissen) Don‘t store Evaluation Events (Auswertungsereignisse nicht speichern) aus.

Nachdem Sie die Felder ausgefüllt haben, wählen Sie Create Project (Projekt erstellen) aus.

Weitere Details finden Sie unter Erstellen eines neuen Projekts.

4. Erstellen Sie nach der Erstellung des Projekts eines Features in diesem Projekt. Nennen Sie das Feature **`showDiscount`**. Erstellen Sie in diesem Feature zwei Varianten des Typs **`Boolean`**. Nennen Sie die erste Variante **`disable`** mit einem Wert von **`False`** und die zweite Variante **`enable`** mit einem Wert von **`True`**.

   Weitere Informationen zum Erstellen eines Features finden Sie unter Hinzufügen eines Features zu einem Projekt.

5. Nachdem Sie mit dem Erstellen des Features fertig sind, erstellen Sie ein Experiment im Projekt. Nennen Sie das Experiment **`pageLoadTime`**.

   Dieses Experiment verwendet eine benutzerdefinierte Metrik namens `pageLoadTime`, die die Seitenladezeit der zu testenden Seite misst. Benutzerdefinierte Metriken für Experimente werden mit Amazon erstellt EventBridge. Weitere Informationen zu EventBridge finden Sie unter Was ist Amazon EventBridge? .

   Um diese benutzerdefinierte Metrik zu erstellen, gehen Sie beim Erstellen des Experiments wie folgt vor:

   - Wählen Sie unter Metrics (Metriken) bei Metric source (Metrikquelle) die Option Custom metrics (Eigene Metriken) aus.
   - Geben Sie bei Metric name den Metriknamen **`pageLoadTime`** ein.
   - Wählen Sie für Goal (Ziel) die Option Decrease (Verringern) aus. Das bedeutet: Ein geringer Wert für diese Metrik kennzeichnet die beste Variante des Features.
   - Geben Sie unter Metric rule (Metrikregel) Folgendes ein:
     - Bei Entity ID (Entitäts-ID) geben Sie **`UserDetails.userId`** ein.
     - Bei Value key (Werteschlüssel) geben Sie **`details.pageLoadTime`** ein.
     - Bei Unit (Einheit) geben Sie **`ms`** ein.
   - Wählen Sie Add metric (Metrik hinzufügen) aus.

Wählen Sie bei Audiences (Zielgruppen) 100% aus, damit alle Benutzer am Experiment teilnehmen. Teilen Sie den Datenverkehr zwischen den Varianten auf jeweils 50 % auf.

Um das Experiment zu erstellen, wählen Sie dann Create Experiment (Experiment erstellen) aus. Nach der Erstellung beginnt es erst, wenn Sie Evidently dazu auffordern, es zu starten.

Schritt 6: Starten Sie das Experiment und testen Sie es CloudWatch offensichtlich

Die letzten Schritte sind das Starten des Experiments und das Starten der Beispiel-App.

Das Experiment aus dem Tutorial starten

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, Evidently aus.
3. Wählen Sie das EvidentlySampleAppProjekt aus.
4. Wechseln Sie zur Registerkarte Experiments (Experimente).
5. Klicken Sie auf die Schaltfläche neben pageLoadTimeund wählen Sie Aktionen, Experiment starten.
6. Wählen Sie einen Zeitpunkt, an dem das Experiment beendet werden soll.
7. Wählen Sie Start Experiment (Experiment starten) aus.

   Das Experiment beginnt sofort.

Starten Sie als Nächstes die Evidenty-Muster-App mit dem folgenden Befehl:

```
npm install -f && npm start
```

Sobald die App gestartet ist, werden Sie einer der beiden zu testenden Feature-Varianten zugewiesen. Bei einer Variante wird „20 % Rabatt“ angezeigt und bei der anderen nicht. Aktualisieren Sie die Seite weiter, um die verschiedenen Varianten zu sehen.

> **Note**
>
> Evidently hat Sticky-Auswertungen. Feature-Auswertung sind deterministisch, d. h. für dieselbe `entityId` und die Funktion erhält ein Benutzer immer dieselbe

Variationszuweisung. Die Zeitvariationszuweisungen ändern sich nur dann, wenn eine Entität zu einer Überschreibung hinzugefügt wird oder der Versuchsverkehr gewählt wird.
Um Ihnen jedoch die Verwendung des Muster-App-Tutorials zu erleichtern, weist Evidently die Feature-Auswertung der Muster-App jedes Mal neu zu, wenn Sie die Seite aktualisieren, sodass Sie beide Varianten erleben können, ohne Überschreibungen hinzufügen zu müssen.

Fehlersuche

Wir empfehlen, `npm` Version 6.14.14 zu verwenden. Wenn Fehler zum Entwickeln oder Starten der Beispiel-App angezeigt werden und Sie eine andere Version von npm verwenden, gehen Sie wie folgt vor.

So installieren Sie die **`npm`**-Version 6.14.14

1. Verwenden Sie einen Browser, um eine Verbindung mit https://nodejs.org/download/release/v14.17.5/ herzustellen.
2. Laden Sie node-v14.17.5.pkg herunter und führen Sie dieses Paket aus, um npm zu installieren.

   Wenn Sie einen `webpack not found`-Fehler sehen, navigieren Sie zum `evidently-sample-shopping-app`-Ordner und versuchen Sie Folgendes:

   a. Löschen Sie `package-lock.json`
   b. Löschen Sie `yarn-lock.json`
   c. Löschen Sie `node_modules`
   d. Löschen Sie die Webpack-Abhängigkeit aus `package.json`
   e. Führen Sie Folgendes aus:

   ```
   npm install -f && npm
   ```

Verwenden Sie CloudWatch RUM

Mit CloudWatch RUM können Sie eine echte Benutzerüberwachung durchführen, um clientseitige Daten über die Leistung Ihrer Webanwendung aus tatsächlichen Benutzersitzungen nahezu in Echtzeit zu sammeln und anzuzeigen. Die Daten, die Sie visualisieren und analysieren können, umfassen Seitenladezeiten, clientseitige Fehler und Benutzerverhalten. Wenn Sie diese Daten

anzeigen, können Sie alle zusammengefasst anzeigen und auch Störungen der Browser und Geräte sehen, die Ihre Kunden verwenden.

Sie können die gesammelten Daten verwenden, um Leistungsprobleme auf Kundenseite schnell zu identifizieren und zu debuggen. CloudWatch RUM hilft Ihnen dabei, Anomalien in der Leistung Ihrer Anwendung zu visualisieren und relevante Debugging-Daten wie Fehlermeldungen, Stack-Traces und Benutzersitzungen zu finden. Sie können RUM auch verwenden, um die Bandbreite der Auswirkungen der Endbenutzer zu verstehen, einschließlich der Anzahl der Benutzer, Geolokalisierungen und Browser.

Endbenutzerdaten, die Sie für CloudWatch RUM sammeln, werden 30 Tage lang aufbewahrt und dann automatisch gelöscht. Wenn Sie die RUM-Ereignisse für einen längeren Zeitraum behalten möchten, können Sie festlegen, dass der App-Monitor Kopien der Ereignisse an die CloudWatch Logs in Ihrem Konto sendet. Anschließend können Sie den Aufbewahrungszeitraum für diese Protokollgruppe anpassen.

Um RUM zu verwenden, erstellen Sie eine App-Überwachung und geben einige Informationen an. RUM generiert einen JavaScript Ausschnitt, den Sie in Ihre Anwendung einfügen können. Der Codeausschnitt zieht den RUM-Webclient-Code ein. Der RUM-Webclient erfasst Daten aus einem Prozentsatz der Benutzersitzungen Ihrer Anwendung, die in einem vorkonfiguriertem Dashboard angezeigt werden. Sie können angeben, aus welchem Prozentsatz der Benutzersitzungen Daten gesammelt werden sollen.

CloudWatch RUM ist in Application Signals integriert, das Ihre Anwendungsdienste, Clients, Synthetics-Kanarien und Serviceabhängigkeiten erkennen und überwachen kann. Verwenden Sie Application Signals, um eine Liste oder eine visuelle Übersicht Ihrer Services zu erhalten, Zustandsmetriken auf der Grundlage Ihrer Servicelevel-Ziele (SLOs) einzusehen und eine detaillierte Darstellung korrelierter X-Ray-Traces für eine detailliertere Fehlerbehebung durchzuführen. Um RUM-Client-Seitenanfragen in Application Signals zu sehen, aktivieren Sie die aktive Nachverfolgung in X-Ray, indem Sie einen Anwendungs-Monitor erstellen oder den RUM-Webclient manuell konfigurieren. Ihre RUM-Clients werden auf der Service-Karte angezeigt, die mit Ihren Services verbunden ist, und auf der Service-Detailseite der Services, die sie aufrufen.

Der RUM-Web-Client ist Open Source. Weitere Informationen finden Sie unter CloudWatch RUM-Webclient.

Leistungsaspekte

In diesem Abschnitt werden die Leistungsaspekte bei der Verwendung von CloudWatch RUM erörtert.

- Auswirkung auf die Ladeleistung — Der CloudWatch RUM-Webclient kann als JavaScript Modul in Ihrer Webanwendung installiert oder asynchron aus einem Content Delivery Network (CDN) in Ihre Webanwendung geladen werden. Der Ladevorgang der Anwendung wird dadurch nicht blockiert. CloudWatch RUM ist so konzipiert, dass es keine spürbaren Auswirkungen auf die Ladezeit der Anwendung gibt.
- Auswirkung auf die Laufzeit — Der RUM-Webclient führt die Verarbeitung durch, um RUM-Daten aufzuzeichnen und an den CloudWatch RUM-Service zu senden. Da Ereignisse selten auftreten und der Verarbeitungsaufwand gering ist, wurde CloudWatch RUM so konzipiert, dass die Leistung der Anwendung nicht nachweisbar beeinträchtigt wird.
- Auswirkungen auf das Netzwerk — Der RUM-Webclient sendet regelmäßig Daten an den CloudWatch RUM-Service. Die Daten werden in regelmäßigen Abständen gesendet, während der Ausführung der Anwendung und auch unmittelbar bevor der Browser die Anwendung entlädt. Daten, die unmittelbar vor dem Entladen des Browsers der Anwendung gesendet werden, werden als Beacons gesendet, die keine erkennbaren Auswirkungen auf die Entladezeit der Anwendung haben.

RUM-Preise

Bei CloudWatch RUM fallen Gebühren für jedes RUM-Ereignis an, das CloudWatch RUM empfängt. Jedes mit dem RUM-Webclient gesammelte Datenelement gilt als RUM-Ereignis. Beispiele für RUM-Ereignisse sind ein Seitenaufruf, ein JavaScript Fehler und ein HTTP-Fehler. Sie haben Optionen dafür, welche Arten von Ereignissen von jedem App-Monitor gesammelt werden.
Sie können Optionen zur Erfassung von Leistungstelemetrieereignissen, JavaScript Fehlern, HTTP-Fehlern und X-Ray-Traces aktivieren oder deaktivieren. Weitere Informationen zu diesen Optionen finden Sie unter Schritt 2: Erstellen Sie einer App-Überwachung und Vom RUM-Webclient gesammelte Informationen CloudWatch . Weitere Informationen zur Preisgestaltung finden Sie unter CloudWatchAmazon-Preise.

Verfügbarkeit in Regionen

CloudWatch RUM ist derzeit in den folgenden Regionen erhältlich:

- USA Ost (Nord-Virginia)
- USA Ost (Ohio)
- USA West (Nordkalifornien)
- USA West (Oregon)
- Afrika (Kapstadt)

- Asien-Pazifik (Jakarta)
- Asien-Pazifik (Mumbai)
- Asien-Pazifik (Hyderabad)
- Asien-Pazifik (Melbourne)
- Asien-Pazifik (Osaka)
- Asia Pacific (Seoul)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)
- Canada (Central)
- Europe (Frankfurt)
- Europa (Irland)
- Europa (London)
- Europa (Milan)
- Europa (Paris)
- Europa (Spain)
- Europa (Stockholm)
- Europa (Zürich)
- Naher Osten (Bahrain)
- Naher Osten (VAE)
- Südamerika (São Paulo)

Themen

- CloudWatch Metriken, die Sie mit CloudWatch RUM sammeln können
- Datenschutz und Datenschutz bei RUM CloudWatch
- Vom RUM-Webclient gesammelte Informationen CloudWatch
- Verwalten Sie Ihre Anwendungen, die CloudWatch RUM verwenden
- CloudWatch RUM-Kontingente
- Problembehandlung bei RUM CloudWatch

IAM-Richtlinien für die Verwendung von RUM CloudWatch

Um CloudWatch RUM vollständig verwalten zu können, müssen Sie als IAM-Benutzer oder als IAM-Rolle angemeldet sein, für die die AmazonCloudWatchFullAccessRUM-IAM-Richtlinie gilt. Darüber hinaus benötigen Sie möglicherweise andere Richtlinien oder Berechtigungen:

- Um einen App-Monitor zu erstellen, der einen neuen Amazon Cognito Cognito-Identitätspool für die Autorisierung erstellt, benötigen Sie die Admin-IAM-Rolle oder die AdministratorAccessIAM-Richtlinie.
- Um einen App-Monitor zu erstellen, der Daten an CloudWatch Logs sendet, müssen Sie bei einer IAM-Rolle oder -Richtlinie angemeldet sein, die über die folgenden Berechtigungen verfügt:

```
{
    "Effect": "Allow",
    "Action": [
        "logs:PutResourcePolicy"
    ],
    "Resource": [
        "*"
    ]
}
```

Anderen Benutzern, die CloudWatch RUM-Daten einsehen, aber keine RUM-Ressourcen erstellen müssen, kann die CloudWatch AmazonCloudWatchReadOnlyAccessRUM-Richtlinie gewährt werden.

Richten Sie eine Anwendung zur Verwendung von CloudWatch RUM ein

Verwenden Sie die Schritte in diesen Abschnitten, um Ihre Anwendung so einzurichten, dass sie mit CloudWatch RUM beginnt, Leistungsdaten aus echten Benutzersitzungen zu sammeln.

Themen

Schritt 1: Autorisieren Sie Ihre Anwendung zum Senden von Daten an AWS

Um CloudWatch RUM verwenden zu können, muss Ihre Anwendung autorisiert sein.

Sie haben drei Möglichkeiten, die Autorisierung einzurichten:

- Lassen Sie CloudWatch RUM einen neuen Amazon Cognito Cognito-Identitätspool für die Anwendung erstellen. Diese Methode erfordert den geringsten Aufwand für die Einrichtung. Dies ist die Standardoption.

  Der Identitätspool enthält eine nicht authentifizierte Identität. Dadurch kann der CloudWatch RUM-Webclient Daten an CloudWatch RUM senden, ohne den Benutzer der Anwendung zu authentifizieren.

  Der Amazon-Cognito-Identitätspool hat eine angehängte IAM-Rolle. Die nicht authentifizierte Identität von Amazon Cognito ermöglicht es dem Webclient, die IAM-Rolle zu übernehmen, die berechtigt ist, Daten an RUM zu senden. CloudWatch
- Verwenden Sie einen vorhandenen Amazon-Cognito-Identitätspool. In diesem Fall müssen Sie auch die IAM-Rolle ändern, die an den Identitätspool angehängt ist. Verwenden Sie diese Option für Identitätspools, die nicht authentifizierte Benutzer unterstützen. Sie können Identitätspools nur aus derselben Region verwenden.
- Verwenden Sie die Authentifizierung von einem vorhandenen Identitätsanbieter, den Sie bereits eingerichtet haben. In diesem Fall müssen Sie Anmeldeinformationen vom Identitätsanbieter abrufen, und Ihre Anwendung muss diese Anmeldeinformationen an den RUM-Webclient weiterleiten.

  Verwenden Sie diese Option für Identitätspools, die nur authentifizierte Benutzer unterstützen.

In den folgenden Abschnitten werden diese Optionen ausführlich erörtert.

CloudWatch RUM erstellt einen neuen Amazon Cognito Cognito-Identitätspool

Dies ist die einfachste Option zum Einrichten. Wenn Sie diese auswählen, sind keine weiteren Einrichtungsschritte erforderlich. Sie benötigen Administratorrechte, um diese Option verwenden zu können. Weitere Informationen finden Sie unter IAM-Richtlinien für die Verwendung von RUM CloudWatch .

Mit dieser Option erstellt CloudWatch RUM die folgenden Ressourcen:

- Einen neuen Amazon-Cognito-Identitätspool
- Eine nicht authentifizierte Amazon-Cognito-Identität. Auf diese Weise kann der RUM-Webclient eine IAM-Rolle übernehmen, ohne den Benutzer der Anwendung zu authentifizieren.
- Die IAM-Rolle, die der RUM-Webclient übernehmen wird. Die an diese Rolle angehängte IAM-Richtlinie ermöglicht die Verwendung der `PutRumEvents`-API mit der App-Überwachungsressource. Mit anderen Worten, es ermöglicht dem RUM-Webclient, Daten an RUM zu senden.

Der RUM-Webclient verwendet die Amazon Cognito Cognito-Identität, um AWS Anmeldeinformationen abzurufen. Die AWS Anmeldeinformationen sind der IAM-Rolle zugeordnet. Die IAM-Rolle ist zur Verwendung `PutRumEvents` mit der AppMonitor Ressource autorisiert.

Amazon Cognito sendet das erforderliche Sicherheitstoken, damit Ihre Anwendung Daten an CloudWatch RUM senden kann. Der von CloudWatch RUM generierte JavaScript Codeausschnitt enthält die folgenden Zeilen, um die Authentifizierung zu ermöglichen.

```
   {
    identityPoolId: [identity pool id],  // e.g., 'us-west-2:EXAMPLE4a-66f6-4114-902a-
EXAMPLEbad7'
   }
);
```

Verwenden Sie einen vorhandenen Amazon-Cognito-Identitätspool.

Wenn Sie sich dafür entscheiden, einen vorhandenen Amazon Cognito Cognito-Identitätspool zu verwenden, geben Sie den Identitätspool an, wenn Sie die Anwendung zu CloudWatch RUM

hinzufügen. Der Pool muss die Aktivierung des Zugriffs auf nicht authentifizierte Identitäten unterstützen. Sie können Identitätspools nur aus derselben Region verwenden.

Sie müssen auch die folgenden Berechtigungen zur IAM-Richtlinie hinzufügen, die an die IAM-Rolle angehängt ist, die mit diesem Identitätspool verknüpft ist.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "rum:PutRumEvents",
            "Resource": "arn:aws:rum:[region]:[accountid]:appmonitor/[app monitor
 name]"
        }
    ]
}
```

Amazon Cognito sendet dann das erforderliche Sicherheitstoken, damit Ihre Anwendung auf CloudWatch RUM zugreifen kann.

Drittanbieter

Wenn Sie sich für die Nutzung einer privaten Authentifizierung von einem Drittanbieter entscheiden, müssen Sie Anmeldeinformationen vom Identitätsanbieter abrufen und an AWS weiterleiten. Am besten kann dies mit einem Sicherheits-Token-Anbieter durchgeführt werden. Sie können jeden Anbieter von Sicherheitstoken verwenden, einschließlich Amazon Cognito mit AWS Security Token Service. Weitere Informationen zu finden Sie AWS STS unter Willkommen bei der AWS Security Token Service API-Referenz.

Wenn Sie Amazon Cognito in diesem Szenario als Token-Anbieter verwenden möchten, können Sie Amazon Cognito so konfigurieren, dass es mit einem Authentifizierungsanbieter zusammenarbeitet. Weitere Informationen erhalten Sie unter Erste Schritte mit Amazon-Cognito-Identitätenpools (Verbundidentitäten).

Nachdem Sie Amazon Cognito für die Zusammenarbeit mit Ihrem Identitätsanbieter konfiguriert haben, müssen Sie wie folgt vorgehen:

- Erstellen Sie eine IAM-Rolle mit den folgenden Berechtigungen. Ihre Anwendung verwendet diese Rolle für den Zugriff auf AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "rum:PutRumEvents",
            "Resource": "arn:aws:rum:[region]:[accountID]:appmonitor/[app monitor
 name]"
        }
    ]
}
```

- Fügen Sie Ihrer Anwendung Folgendes hinzu, damit sie die Anmeldeinformationen von Ihrem Anbieter an CloudWatch RUM weiterleitet. Fügen Sie die Zeile so ein, dass sie ausgeführt wird, nachdem sich ein Benutzer bei Ihrer Anwendung angemeldet hat und die Anwendung die Anmeldeinformationen für den Zugriff auf AWS erhalten hat.

```
cwr('setAwsCredentials', {/* Credentials or CredentialProvider */});
```

Weitere Informationen zu Anmeldeinformationsanbietern im AWS JavaScript SDK finden Sie unter Einrichten von Anmeldeinformationen in einem Webbrowser im v3-Entwicklerhandbuch für SDK for JavaScript, Einrichten von Anmeldeinformationen in einem Webbrowser im v2-Entwicklerhandbuch für SDK for JavaScript, und @aws -sdk/credentials al-providers.

Sie können auch das SDK für den CloudWatch RUM-Webclient verwenden, um die Webclient-Authentifizierungsmethoden zu konfigurieren. Weitere Informationen zum Webclient-SDK finden Sie unter CloudWatch RUM-Webclient-SDK.

Schritt 2: Erstellen Sie einer App-Überwachung

Um mit der Verwendung von CloudWatch RUM mit Ihrer Anwendung zu beginnen, erstellen Sie einen App-Monitor. Wenn der App-Monitor erstellt wird, generiert RUM einen JavaScript Ausschnitt, den Sie in Ihre Anwendung einfügen können. Der Codeausschnitt zieht den RUM-Webclient-Code ein. Der RUM-Webclient erfasst Daten aus einem Prozentsatz der Benutzersitzungen Ihrer Anwendung und sendet sie an RUM.

App-Überwachung erstellen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

2. Wählen Sie im Navigationsbereich Application Signals, RUM aus.
3. Wählen Sie Add app monitor (App-Überwachung hinzufügen) aus.
4. Geben Sie die Informationen und Einstellungen für Ihre Anwendung ein:

   - Geben Sie unter Name des App-Monitors einen Namen ein, der zur Identifizierung dieses App-Monitors in der CloudWatch RUM-Konsole verwendet werden soll.
   - Geben Sie für Domain der Anwendung den Domainnamen der obersten Ebene ein, unter der Ihre Anwendung über Verwaltungsberechtigung verfügt. Dies muss ein URL-Domainformat sein.

     Wählen Sie Include sub domains (Untergeordnete Domains einbeziehen) aus, damit die App-Überwachung auch Daten von allen untergeordneten Domains unter der Domain der obersten Ebene sammelt.
5. Geben Sie für Configure RUM data collection (RUM-Datensammlung konfigurieren) an, ob die App-Überwachung alle folgenden Aspekte sammeln soll:

   - Performance-Telemetrie – Sammelt Informationen über Seitenlade- und Ressourcenladezeiten
   - JavaScript Fehler — Sammelt Informationen über unbehandelte JavaScript Fehler, die von Ihrer Anwendung verursacht wurden
   - HTTP-Fehler – Sammelt Informationen über HTTP-Fehler, die von Ihrer Anwendung ausgegeben werden

   Wenn Sie diese Optionen auswählen, erhalten Sie mehr Informationen zu Ihrer Anwendung, es werden jedoch auch mehr CloudWatch RUM-Ereignisse generiert, sodass mehr Gebühren anfallen.

   Wenn Sie keine dieser Optionen auswählen, sammelt die App-Überwachung weiterhin Sitzungsstartereignisse und Seiten-IDs, sodass Sie sehen, wie viele Benutzer Ihre Anwendung verwenden. Sie können außerdem Störungen nach Betriebssystemtyp und -version, Browsertyp und -version, Gerätetyp und Speicherort anzeigen.
6. Wählen Sie Diese Option aktivieren, damit der CloudWatch RUM-Webclient Cookies setzen kann, wenn Sie Benutzer-IDs und Sitzungs-IDs aus Stichproben von Benutzersitzungen sammeln möchten. Die Benutzer-IDs werden nach dem Zufallsprinzip von RUM generiert. Weitere Informationen finden Sie unter CloudWatch RUM-Webclient-Cookies (oder ähnliche Technologien).

7. Geben Sie für Session samples (Sitzungsbeispiele) den Prozentsatz der Benutzersitzungen ein, die zum Sammeln von RUM-Daten verwendet werden. Der Standardwert ist 100 %. Wenn Sie diese Zahl reduzieren, erhalten Sie weniger Daten und die Gebühren sinken. Weitere Informationen zu den Preisen für RUM erhalten Sie unter RUM – Preise.
8. Endbenutzerdaten, die Sie für CloudWatch RUM sammeln, werden 30 Tage lang aufbewahrt und dann gelöscht. Wenn Sie Kopien von RUM-Ereignissen in CloudWatch Logs speichern und konfigurieren möchten, wie lange diese Kopien aufbewahrt werden sollen, wählen Sie Diese Option aktivieren, um Ihre Anwendungstelemetriedaten in Ihrem CloudWatch Logs-Konto unter Datenspeicher zu speichern. Standardmäßig speichert die Protokollgruppe CloudWatch Logs die Daten 30 Tage lang. Sie können den Aufbewahrungszeitraum in der CloudWatch Logs-Konsole anpassen.
9. Geben Sie für Authorization (Autorisierung) an, ob Sie einen neuen oder vorhandenen Amazon-Cognito-Identitätspool oder einen anderen Identitätsanbieter verwenden möchten. Das Erstellen eines neuen Identitätspools ist die einfachste Option, die keine anderen Einrichtungsschritte erfordert. Weitere Informationen finden Sie unter Schritt 1: Autorisieren Sie Ihre Anwendung zum Senden von Daten an AWS.

   Das Erstellen eines neuen Amazon-Cognito-Identitätspools erfordert Administratorberechtigungen. Weitere Informationen finden Sie unter IAM-Richtlinien für die Verwendung von RUM CloudWatch .
10. (Optional) Wenn Sie den RUM-Codeausschnitt zu Ihrer Anwendung hinzufügen, fügt der Webclient das JavaScript Tag zur Überwachung der Nutzung standardmäßig in den HTML-Code aller Seiten Ihrer Anwendung ein. Um dies zu ändern, wählen Sie Configure pages (Seiten konfigurieren) und anschließend entweder Include only these pages (Nur diese Seiten einschließen) oder Exclude these pages (Diese Seiten ausschließen) aus. Geben Sie dann die Seiten an, die ein- oder ausgeschlossen werden sollen. Um eine Seite anzugeben, die ein- oder ausgeschlossen werden soll, geben Sie die vollständigen URLs ein. Um zusätzliche Seiten anzugeben, wählen Sie Add URL (URL hinzufügen) aus.
11. Um die AWS X-Ray Ablaufverfolgung der Benutzersitzungen zu aktivieren, die vom App Monitor erfasst werden, wählen Sie Aktives Tracing und dann Meinen Service verfolgen mit aus. AWS X-Ray

    Wenn Sie diese Auswahl treffen, werden `XMLHttpRequest`- und `fetch`-Anforderungen, die während Benutzersitzungen, die von der App-Überwachung im Rahmen einer Stichprobe ausgewählt werden, nachverfolgt. Sie sehen dann Ablaufverfolgungen und Segmente aus diesen Benutzersitzungen im RUM-Dashboard sowie die X-Ray-Trace-Karte und die Trace-

Detailseiten. Diese Benutzersitzungen werden auch als Client-Seiten in Application Signals angezeigt, nachdem Sie sie für Ihre Anwendung aktiviert haben.

Indem Sie zusätzliche Konfigurationsänderungen am CloudWatch RUM-Webclient vornehmen, können Sie HTTP-Anfragen einen X-Ray-Trace-Header hinzufügen, um die end-to-end Nachverfolgung von Benutzersitzungen bis hin zu nachgeschalteten AWS Managed Services zu ermöglichen. Weitere Informationen finden Sie unter X-Ray end-to-end Tracing aktivieren.

12. (Optional) Um Tags zur App-Überwachung hinzuzufügen, wählen Sie Tags (Tags), Add new tag (Neues Tag hinzufügen) aus.

    Geben Sie für Key (Schlüssel) einen Namen für das Tag ein. Sie können einen optionalen Wert für das Tag unter Value (Wert) hinzufügen.

    (Optional) Zum Hinzufügen eines weiteren Tags wählen Sie Add new tag (Neues Tag hinzufügen) erneut aus.

    Weitere Informationen finden Sie unter Ressourcen taggen AWS.

13. Wählen Sie Add app monitor (App-Überwachung hinzufügen) aus.
14. Im Abschnitt Sample code (Beispiel-Code) können Sie den Codeausschnitt kopieren, um ihn in Ihre Anwendung einzufügen. Wir empfehlen, dass Sie JavaScriptoder wählen TypeScriptund NPM verwenden, CloudWatch um den RUM-Webclient als JavaScript Modul zu installieren.

    Alternativ können Sie HTML wählen, um ein Content Delivery Network (CDN) zur Installation des CloudWatch RUM-Webclients zu verwenden. Der Nachteil der Verwendung eines CDN ist, dass der Webclient oft von Werbeblockern blockiert wird.

15. Wählen Sie Copy (Kopieren) oder Download (Herunterladen), und klicken Sie dann auf Done (Fertig).

(Optional) Schritt 3: Ändern Sie den Codeausschnitt manuell, um den CloudWatch RUM-Webclient zu konfigurieren

Sie können den Codeausschnitt ändern, bevor Sie ihn in Ihre Anwendung einfügen, um mehrere Optionen zu aktivieren oder zu deaktivieren. Weitere Informationen finden Sie in der CloudWatch RUM-Webclient-Dokumentation.

Es gibt drei Konfigurationsoptionen, die Sie unbedingt beachten sollten, wie in diesen Abschnitten beschrieben.

Verhindern der Erfassung von Ressourcen-URLs, die möglicherweise personenbezogene Daten enthalten

Standardmäßig ist der CloudWatch RUM-Webclient so konfiguriert, dass er die URLs der von der Anwendung heruntergeladenen Ressourcen aufzeichnet. Zu diesen Ressourcen gehören HTML-Dateien, Bilder, JavaScript CSS-Dateien, Dateien usw. Für einige Anwendungen können URLs persönlich identifizierbare Informationen (PII) enthalten.

Wenn dies bei Ihrer Anwendung der Fall ist, empfehlen wir dringend, die Erfassung von Ressourcen-URLs zu deaktivieren, indem Sie `recordResourceUrl: false` in der Codeausschnitt-Konfiguration festlegen, bevor Sie es in Ihre Anwendung einfügen.

Manuelle Aufzeichnung von Seitenaufrufen

Standardmäßig zeichnet der Webclient Seitenaufrufe auf, wenn die Seite zum ersten Mal geladen wird und wenn die Browser-Verlaufs-API aufgerufen wird. Die Standardseiten-ID ist `window.location.pathname`. In manchen Fällen möchten Sie dieses Verhalten jedoch außer Kraft setzen und die Anwendung so instrumentieren, dass die Seitenaufrufe programmatisch aufgezeichnet werden. Auf diese Weise haben Sie die Kontrolle über die Seiten-ID und den Zeitpunkt ihrer Aufzeichnung. Nehmen wir zum Beispiel eine Webanwendung, die eine URI mit einer variablen ID hat, wie `/entity/123` oder `/entity/456`. Standardmäßig generiert CloudWatch RUM für jeden URI ein Seitenaufrufereignis mit einer eindeutigen Seiten-ID, die dem Pfadnamen entspricht. Möglicherweise möchten Sie sie jedoch stattdessen nach derselben Seiten-ID gruppieren. Deaktivieren Sie dazu in der `disableAutoPageView`-Konfiguration die Automatisierung der Seitenansicht des Webclients und verwenden Sie den Befehl `recordPageView` zum Festlegen der gewünschten Seiten-ID. Weitere Informationen finden Sie unter Anwendungsspezifische Konfigurationen unter. GitHub

Beispiel für eingebettetes Skript:

```
cwr('recordPageView', { pageId: 'entityPageId' });
```

JavaScript Beispiel für ein Modul:

```
awsRum.recordPageView({ pageId: 'entityPageId' });
```

X-Ray end-to-end Tracing aktivieren

Wenn Sie die App-Überwachung erstellen, wählen Sie Meinen Service nachverfolgen mit AWS X-Ray aus. Dies ermöglicht das Nachverfolgen von `XMLHttpRequest`- und `fetch`-Anforderungen,

die während Benutzersitzungen gestellt werden, die von der App-Überwachung als Stichprobe ausgewählt werden. Sie können dann die Traces dieser HTTP-Anfragen im CloudWatch RUM-Dashboard sowie auf den Seiten X-Ray Trace Map und Trace-Details sehen.

Standardmäßig sind diese clientseitigen Ablaufverfolgungen nicht mit nachgelagerten serverseitigen Ablaufverfolgungen verbunden. Um clientseitige Traces mit serverseitigen Traces zu verbinden und end-to-end Tracing zu aktivieren, setzen Sie die `addXRayTraceIdHeader` Option `true` im Webclient auf. Dadurch fügt der CloudWatch RUM-Webclient HTTP-Anfragen einen X-Ray-Trace-Header hinzu.

Der folgende Codeblock zeigt ein Beispiel für das Hinzufügen clientseitiger Ablaufverfolgungen. Einige Konfigurationsoptionen werden aus diesem Beispiel zu Gunsten der Lesbarkeit weggelassen.

```
<script>
    (function(n,i,v,r,s,c,u,x,z){...})(
        'cwr',
        '00000000-0000-0000-0000-000000000000',
        '1.0.0',
        'us-west-2',
        'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
        {
            enableXRay: true,
            telemetries: [
                'errors',
                'performance',
                [ 'http', { addXRayTraceIdHeader: true } ]
            ]
        }
    );
</script>
```

> **Warning**
>
> Wenn Sie den CloudWatch RUM-Webclient so konfigurieren, dass er HTTP-Anfragen einen X-Ray-Trace-Header hinzufügt, kann dies dazu führen, dass Cross-Origin Resource Sharing (CORS) fehlschlägt oder die Signatur der Anfrage ungültig wird, wenn die Anfrage mit Sigv4 signiert ist. Weitere Informationen finden Sie in der CloudWatch RUM-Webclient-Dokumentation. Wir empfehlen dringend, Ihre Anwendung zu testen, bevor Sie einen clientseitigen X-Ray-Ablaufverfolgungs-Header in einer Produktionsumgebung hinzufügen.

Weitere Informationen finden Sie in der CloudWatch RUM-Webclient-Dokumentation

Schritt 4: Fügen Sie den Codeausschnitt in Ihre Anwendung ein

Als Nächstes fügen Sie den Codeausschnitt, den Sie im vorherigen Abschnitt erstellt haben, in Ihre Anwendung ein.

> **Warning**
>
> Der Webclient, der durch den Codeausschnitt heruntergeladen und konfiguriert wurde, verwendet Cookies (oder ähnliche Technologien), um Ihnen bei der Erfassung von Endbenutzerdaten zu helfen. Bevor Sie den Codeausschnitt einfügen, sehen Sie sich Filtern nach Metadatenattributen in der Konsole an.

Wenn Sie den zuvor generierten Codeausschnitt nicht haben, finden Sie ihn, indem Sie den Anweisungen in Wie finde ich einen Codeausschnitt, den ich bereits generiert habe? folgen.

Um den CloudWatch RUM-Codeausschnitt in Ihre Anwendung einzufügen

1. Fügen Sie den Codeausschnitt, den Sie im vorherigen Abschnitt kopiert oder heruntergeladen haben, in das <head>-Element Ihrer Anwendung ein. Fügen Sie ihn vor dem <body>-Element oder anderen <script>-Tags ein.

   Beispiel für einen generierten Codeausschnitt:

```
<script>
(function (n, i, v, r, s, c, x, z) {
    x = window.AwsRumClient = {q: [], n: n, i: i, v: v, r: r, c: c};
    window[n] = function (c, p) {
        x.q.push({c: c, p: p});
    };
    z = document.createElement('script');
    z.async = true;
    z.src = s;
    document.head.insertBefore(z, document.getElementsByTagName('script')[0]);
})('cwr',
    '194a1c89-87d8-41a3-9d1b-5c5cd3dafbd0',
    '1.0.0',
    'us-east-2',
    'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
```

```
    {
        sessionSampleRate: 1,
        identityPoolId: "us-east-2:c90ef0ac-e3b8-4d1a-b313-7e73cfd21443",
        endpoint: "https://dataplane.rum.us-east-2.amazonaws.com",
        telemetries: ["performance", "errors", "http"],
        allowCookies: true,
        enableXRay: false
    });
</script>
```

2. Wenn es sich bei Ihrer Anwendung um eine mehrseitige Webanwendung handelt, müssen Sie Schritt 1 für jede HTML-Seite wiederholen, die in die Datenerfassung aufgenommen werden soll.

Schritt 5: Testen Sie die Einrichtung der App-Überwachung, indem Sie Benutzerereignisse generieren

Nachdem Sie den Codeausschnitt eingefügt haben und Ihre aktualisierte Anwendung ausgeführt wird, können Sie sie testen, indem Sie Benutzerereignisse manuell generieren. Wir empfehlen für den Test Folgendes: Für diesen Test fallen die üblichen CloudWatch RUM-Gebühren an.

- Navigieren Sie zwischen Seiten in Ihrer Webanwendung.
- Erstellen Sie mehrere Benutzersitzungen mit verschiedenen Browsern und Geräten.
- Stellen Sie Anforderungen.
- JavaScript Fehler verursachen.

Nachdem Sie einige Ereignisse generiert haben, können Sie sie im CloudWatch RUM-Dashboard anzeigen. Weitere Informationen finden Sie unter Das CloudWatch RUM-Dashboard anzeigen.

Es kann bis zu 15 Minuten dauern, bis Daten aus Benutzersitzungen im Dashboard angezeigt werden.

Wenn 15 Minuten nach dem Generieren von Ereignissen in der Anwendung keine Daten angezeigt werden, ziehen Sie Problembehandlung bei RUM CloudWatch zurate.

Konfiguration des CloudWatch RUM-Webclients

Ihre Anwendungen können eines der von RUM generierten Codefragmente verwenden, um den CloudWatch CloudWatch RUM-Webclient zu installieren. Die generierten Codefragmente

unterstützen zwei Installationsmethoden: als JavaScript Modul über NPM oder über ein Content Delivery Network (CDN). Um die beste Leistung zu erzielen, empfehlen wir die NPM-Installationsmethode. Weitere Informationen zur Verwendung dieser Methode finden Sie unter Als Modul installieren. JavaScript

Wenn Sie die CDN-Installationsoption verwenden, blockieren Werbeblocker möglicherweise das von RUM bereitgestellte Standard-CDN. CloudWatch Damit wird die Anwendungsüberwachung für Benutzer deaktiviert, die Werbeblocker installiert haben. Aus diesem Grund empfehlen wir, das Standard-CDN nur für das erste Onboarding mit RUM zu verwenden. CloudWatch Weitere Informationen zur Behebung dieses Problems finden Sie unter Instrumentierung dieser Anwendung.

Der Codeausschnitt befindet sich im <head>-Tag einer HTML-Datei und installiert den Webclient, indem es den Webclient herunterlädt und ihn dann für die zu überwachende Anwendung konfiguriert. Der Ausschnitt ist eine selbstausführende Funktion, die wie folgt aussieht. In diesem Beispiel wurde ein Großteil der Funktion des Ausschnitts zu Gunsten der Lesbarkeit weggelassen.

```
<script>
(function(n,i,v,r,s,c,u,x,z){...})(
'cwr',
'00000000-0000-0000-0000-000000000000',
'1.0.0',
'us-west-2',
'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
{ /* Configuration Options Here */ }
);
<script>
```

Argumente

Der Codeausschnitt akzeptiert sechs Argumente:

- Ein Namespace zum Ausführen von Befehlen auf dem Webclient, wie `'cwr'`
- Die ID der App-Überwachung, wie z. B. `'00000000-0000-0000-0000-000000000000'`
- Die Anwendungsversion wie z. B. `'1.0.0'`
- Die AWS Region des App-Monitors, z. B. `'us-west-2'`
- Die URL des Web-Clients wie z. B. `'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js'`
- Anwendungsspezifische Konfigurationsoptionen. Weitere Informationen finden Sie im folgenden Abschnitt.

Ignorieren von Fehlern

Der CloudWatch RUM-Webclient überwacht alle Arten von Fehlern, die in Ihren Anwendungen auftreten. Wenn Ihre Anwendung JavaScript Fehler ausgibt, die Sie nicht im CloudWatch RUM-Dashboard anzeigen möchten, können Sie den RUM-Webclient so konfigurieren, dass diese Fehler herausgefiltert werden, sodass Sie nur die relevanten Fehlerereignisse im CloudWatch RUM-Dashboard sehen. CloudWatch Sie könnten sich beispielsweise dafür entscheiden, einige JavaScript Fehler nicht im Dashboard anzuzeigen, weil Sie bereits eine Lösung für sie gefunden haben und die Menge dieser Fehler andere Fehler maskiert. Unter Umständen möchten Sie auch Fehler ignorieren, die Sie nicht beheben können, da sie mit einer Bibliothek eines Drittanbieters zusammenhängen.

Weitere Informationen zur Instrumentierung des Webclients zum Herausfiltern bestimmter JavaScript Fehler finden Sie im Beispiel unter Fehler in der Github-Dokumentation für den Webclient.

Konfigurationsoptionen

Informationen zu den für den CloudWatch RUM-Webclient verfügbaren Konfigurationsoptionen finden Sie in der CloudWatch RUM-Webclient-Dokumentation

Regionalisierung

In diesem Abschnitt werden Strategien für die Verwendung von CloudWatch RUM mit Anwendungen in verschiedenen Regionen beschrieben.

Meine Webanwendung wird in mehreren AWS Regionen bereitgestellt

Wenn Ihre Webanwendung in mehreren AWS Regionen bereitgestellt wird, haben Sie drei Optionen:

- Stellen Sie einen App-Monitor in einer Region, in einem Konto, für alle Regionen bereit.
- Stellen Sie separate App-Monitore für jede Region in eindeutigen Konten bereit.
- Stellen Sie separate App-Monitore für jede Region bereit, alles in einem Konto.

Der Vorteil der Verwendung eines App-Monitors besteht darin, dass alle Daten in einer Visualisierung zentralisiert werden und alle Protokolle in dieselbe Protokollgruppe unter CloudWatch Logs geschrieben werden. Bei einem einzelnen App-Monitor gibt es eine geringe zusätzliche Latenzzeit für Anfragen und einen einzigen Fehlerpunkt.

Die Verwendung mehrerer App-Monitore beseitigt zwar den einzelnen Fehlerpunkt, verhindert aber, dass alle Daten in einer Visualisierung zusammengefasst werden.

CloudWatch RUM wurde in einigen Regionen, in denen meine Anwendung bereitgestellt wird, nicht gestartet

CloudWatch RUM wird in vielen Regionen eingeführt und hat eine breite geografische Abdeckung. Wenn Sie CloudWatch RUM in den Regionen einrichten, in denen es verfügbar ist, können Sie die Vorteile nutzen. Endbenutzer können sich überall befinden und ihre Sitzungen werden trotzdem erfasst, wenn Sie einen App-Monitor in der Region eingerichtet haben, mit der sie sich verbinden.

CloudWatch RUM ist jedoch noch nicht in AWS GovCloud (USA-Ost), AWS GovCloud (US-West) oder anderen Regionen Chinas eingeführt. Sie können aus diesen Regionen keine Daten an CloudWatch RUM senden.

Verwenden von Seitengruppen

Mithilfe von Seitengruppen können Sie verschiedene Seiten in Ihrer Anwendung miteinander verknüpfen, um aggregierte Analysen für Seitengruppen anzuzeigen. So können Sie z. B. die aggregierten Seitenladezeiten Ihrer gesamten Zielseiten anzeigen.

Sie ordnen Seiten in Seitengruppen ein, indem Sie zu Seitenaufrufereignissen im CloudWatch RUM-Webclient ein oder mehrere Tags hinzufügen. In den folgenden Beispielen wird die Seite `/home` in den Seitengruppen `en` und `landing` platziert:

Beispiel für eingebettetes Skript

```
cwr('recordPageView', { pageId: '/home', pageTags: ['en', 'landing']});
```

JavaScript Beispiel für ein Modul

```
awsRum.recordPageView({ pageId: '/home', pageTags: ['en', 'landing']});
```

> **Note**
>
> Seitengruppen sind dazu gedacht, Analysen über verschiedene Seiten hinweg zu aggregieren. Informationen darüber, wie Sie `pageIds` für Ihre Anwendung definieren und manipulieren können, finden Sie im Abschnitt Manuelles Aufzeichnen von Seitenaufrufen in (Optional) Schritt 3: Ändern Sie den Codeausschnitt manuell, um den CloudWatch RUM-Webclient zu konfigurieren.

Benutzerdefinierte Metadaten angeben

CloudWatch RUM fügt jedem Ereignis zusätzliche Daten als Metadaten hinzu. Ereignismetadaten bestehen aus Attributen in Form von Schlüssel-Wert-Paaren. Sie können diese Attribute verwenden, um Ereignisse in der CloudWatch RUM-Konsole zu suchen oder zu filtern. Standardmäßig erstellt CloudWatch RUM einige Metadaten für Sie. Weitere Informationen zu Standard-Metadaten erhalten Sie unter RUM-Ereignismetadaten.

Sie können den CloudWatch RUM-Webclient auch verwenden, um benutzerdefinierte Metadaten zu CloudWatch RUM-Ereignissen hinzuzufügen. Die benutzerdefinierten Metadaten können Sitzungsattribute und Seitenattribute enthalten.

Um benutzerdefinierte Metadaten hinzuzufügen, müssen Sie Version 1.10.0 oder höher des CloudWatch RUM-Webclients verwenden.

Anforderungen und Syntax

Jedes Ereignis kann bis zu 10 benutzerdefinierte Attribute in den Metadaten enthalten. Die Syntaxanforderungen für benutzerdefinierte Attribute lauten wie folgt:

- Schlüssel
  - Maximal 128 Zeichen
  - Kann alphanumerische Zeichen, Doppelpunkte (:) und Unterstriche (_) enthalten
  - Darf nicht mit `aws:` beginnen.
  - Darf nicht vollständig aus einem der reservierten Schlüsselwörter bestehen, die im folgenden Abschnitt aufgeführt sind. Kann diese Schlüsselwörter als Teil eines längeren Schlüsselnamens verwenden.
- Werte
  - Maximal 256 Zeichen
  - Müssen Zeichenfolgen, Zahlen oder boolesche Werte sein

Reservierte Schlüsselwörter

Sie können die folgenden reservierten Schlüsselnamen nicht als vollständige Schlüsselnamen verwenden. Sie können die folgenden Schlüsselwörter als Teil eines längeren Schlüsselnamens, wie z. B. `applicationVersion`, verwenden.

- `browserLanguage`

- `browserName`
- `browserVersion`
- `countryCode`
- `deviceType`
- `domain`
- `interaction`
- `osName`
- `osVersion`
- `pageId`
- `pageTags`
- `pageTitle`
- `pageUrl`
- `parentPageId`
- `platformType`
- `referrerUrl`
- `subdivisionCode`
- `title`
- `url`
- `version`

> **Note**
>
> CloudWatch RUM entfernt benutzerdefinierte Attribute aus RUM-Ereignissen, wenn ein Attribut einen ungültigen Schlüssel oder Wert enthält oder wenn das Limit von 10 benutzerdefinierten Attributen pro Ereignis bereits erreicht wurde.

Sitzungsattribute hinzufügen

Wenn Sie benutzerdefinierte Sitzungsattribute konfigurieren, werden sie allen Ereignissen in einer Sitzung hinzugefügt. Sie konfigurieren Sitzungsattribute entweder während der Initialisierung des CloudWatch RUM-Webclients oder zur Laufzeit mithilfe des `addSessionAttributes` Befehls.

Beispielsweise können Sie die Version Ihrer Anwendung als Sitzungsattribut hinzufügen. Anschließend können Sie in der CloudWatch RUM-Konsole Fehler nach Version filtern, um herauszufinden, ob eine erhöhte Fehlerrate mit einer bestimmten Version Ihrer Anwendung zusammenhängt.

Hinzufügen eines Sitzungsattributs bei der Initialisierung, NPM-Beispiel

Der fett gedruckte Codeabschnitt fügt das Sitzungsattribut hinzu.

```
import { AwsRum, AwsRumConfig } from 'aws-rum-web';

try {
  const config: AwsRumConfig = {
    allowCookies: true,
    endpoint: "https://dataplane.rum.us-west-2.amazonaws.com",
    guestRoleArn: "arn:aws:iam::000000000000:role/RUM-Monitor-us-
west-2-000000000000-00xx-Unauth",
    identityPoolId: "us-west-2:00000000-0000-0000-0000-000000000000",
    sessionSampleRate: 1,
    telemetries: ['errors', 'performance'],
    sessionAttributes: {
        applicationVersion: "1.3.8"
    }
  };

  const APPLICATION_ID: string = '00000000-0000-0000-0000-000000000000';
  const APPLICATION_VERSION: string = '1.0.0';
  const APPLICATION_REGION: string = 'us-west-2';

  const awsRum: AwsRum = new AwsRum(
    APPLICATION_ID,
    APPLICATION_VERSION,
    APPLICATION_REGION,
    config
  );
} catch (error) {
  // Ignore errors thrown during CloudWatch RUM web client initialization
}
```

Hinzufügen eines Sitzungsattributs zur Laufzeit, NPM-Beispiel

```
awsRum.addSessionAttributes({
```

```
    applicationVersion: "1.3.8"
})
```

Hinzufügen eines Sitzungsattributs bei der Initialisierung, Beispiel für eingebettetes Skript

Der fett gedruckte Codeabschnitt fügt das Sitzungsattribut hinzu.

```
<script>
    (function(n,i,v,r,s,c,u,x,z){...})(
        'cwr',
        '00000000-0000-0000-0000-000000000000',
        '1.0.0',
        'us-west-2',
        'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
        {
            sessionSampleRate:1,
            guestRoleArn:'arn:aws:iam::000000000000:role/RUM-Monitor-us-
west-2-000000000000-00xx-Unauth',
            identityPoolId:'us-west-2:00000000-0000-0000-0000-000000000000',
            endpoint:'https://dataplane.rum.us-west-2.amazonaws.com',
            telemetries:['errors','http','performance'],
            allowCookies:true,
            sessionAttributes: {
                applicationVersion: "1.3.8"
            }
        }
    );
</script>
```

Hinzufügen eines Sitzungsattributs zur Laufzeit, Beispiel für eingebettetes Skript

```
<script>
    function addSessionAttribute() {
        cwr('addSessionAttributes', {
            applicationVersion: "1.3.8"
        })
    }

</script>
```

Seitenattribute hinzufügen

Wenn Sie benutzerdefinierte Seitenattribute konfigurieren, werden sie allen Ereignissen in einer Sitzung hinzugefügt. Sie konfigurieren Seitenattribute entweder während der Initialisierung des CloudWatch RUM-Webclients oder zur Laufzeit mithilfe des `recordPageView` Befehls.

Beispielsweise können Sie Ihre Seitenvorlage als Seitenattribut hinzufügen. Anschließend können Sie in der CloudWatch RUM-Konsole Fehler nach Seitenvorlagen filtern, um herauszufinden, ob eine erhöhte Fehlerrate mit einer bestimmten Seitenvorlage Ihrer Anwendung verknüpft ist.

Hinzufügen eines Seitenattributs bei der Initialisierung, NPM-Beispiel

Der fett gedruckte Codeabschnitt fügt das Seitenattribut hinzu.

```
const awsRum: AwsRum = new AwsRum(
    APPLICATION_ID,
    APPLICATION_VERSION,
    APPLICATION_REGION,
    { disableAutoPageView:  true // optional }
);
awsRum.recordPageView({
    pageId:'/home',
    pageAttributes: {
      template: 'artStudio'
    }
});
const credentialProvider = new CustomCredentialProvider();
if(awsCreds) awsRum.setAwsCredentials(credentialProvider);
```

Hinzufügen eines Seitenattributs zur Laufzeit, NPM-Beispiel

```
awsRum.recordPageView({
    pageId: '/home',
    pageAttributes: {
        template: 'artStudio'
    }
});
```

Hinzufügen eines Seitenattributs bei der Initialisierung, Beispiel für eingebettetes Skript

Der fett gedruckte Codeabschnitt fügt das Seitenattribut hinzu.

```
<script>
    (function(n,i,v,r,s,c,u,x,z){...})(
        'cwr',
        '00000000-0000-0000-0000-000000000000',
        '1.0.0',
        'us-west-2',
        'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
        {
            disableAutoPageView: true //optional
        }
    );
    cwr('recordPageView', {
       pageId: '/home',
       pageAttributes: {
           template: 'artStudio'
       }
    });
    const awsCreds = localStorage.getItem('customAwsCreds');
    if(awsCreds) cwr('setAwsCredentials', awsCreds)
</script>
```

Hinzufügen eines Seitenattributs zur Laufzeit, Beispiel für eingebettetes Skript

```
<script>
    function recordPageView() {
        cwr('recordPageView', {
            pageId: '/home',
            pageAttributes: {
                template: 'artStudio'
            }
        });
    }
</script>
```

Filtern nach Metadatenattributen in der Konsole

Verwenden Sie die Suchleiste, um die Visualisierungen in der CloudWatch RUM-Konsole mit einem beliebigen integrierten oder benutzerdefinierten Metadatenattribut zu filtern. In der Suchleiste können Sie bis zu 20 Filterbegriffe in der Form key=value (Schlüssel=Wert) angeben, die auf die Visualisierungen angewendet werden sollen. Um beispielsweise Daten nur für den Chrome-Browser zu filtern, könnten Sie den Filterbegriff browserName=Chrome (Browsername=Chrome) hinzufügen.

Standardmäßig ruft die CloudWatch RUM-Konsole die 100 häufigsten Attributschlüssel und -werte ab, die in der Dropdownliste in der Suchleiste angezeigt werden. Um weitere Metadatenattribute als Filterbegriffe hinzuzufügen, geben Sie den vollständigen Attributschlüssel und -wert in die Suchleiste ein.

Ein Filter kann bis zu 20 Filterbegriffe enthalten und Sie können bis zu 20 Filter pro Anwendungsmonitor speichern. Wenn Sie einen Filter speichern, wird er in der Dropdownliste Saved filters (Gespeicherte Filter). Sie können einen gespeicherten Filter auch löschen.

Benutzerdefinierte Ereignisse senden

CloudWatch RUM zeichnet die unter aufgelisteten Ereignisse auf und nimmt sie aufVom RUM-Webclient gesammelte Informationen CloudWatch . Wenn Sie Version 1.12.0 oder höher des CloudWatch RUM-Webclients verwenden, können Sie zusätzliche benutzerdefinierte Ereignisse definieren, aufzeichnen und senden. Sie definieren den Namen des Ereignistyps und die zu sendenden Daten für jeden Ereignistyp, den Sie definieren. Jede benutzerdefinierte Ereignis-Nutzlast kann bis zu 6 KB groß sein.

Benutzerdefinierte Ereignisse werden nur erfasst, wenn der Anwendungsmonitor benutzerdefinierte Ereignisse aktiviert hat. Verwenden Sie die CloudWatch RUM-Konsole oder die API, um die Konfigurationseinstellungen Ihres App Monitors zu aktualisieren. UpdateAppMonitor

Nachdem Sie benutzerdefinierte Ereignisse aktiviert und anschließend benutzerdefinierte Ereignisse definiert und gesendet haben, können Sie nach ihnen suchen. Verwenden Sie die Registerkarte Ereignisse in der CloudWatch RUM-Konsole, um nach ihnen zu suchen. Suchen Sie anhand des Ereignistyps.

Anforderungen und Syntax

Benutzerdefinierte Ereignisse bestehen aus einem Ereignistyp und Ereignisdetails. Die Anforderungen für diese sind Folgende:

- Ereignistyp
  - Dies kann entweder der type (Typ) oder der name (Name) Ihres Ereignisses sein. Beispielsweise JsErrorhat der in CloudWatch RUM integrierte Ereignistyp namens den Ereignistyp`com.amazon.rum.js_error_event`.
  - Muss zwischen 1 und 256 Zeichen lang sein.
  - Kann eine Kombination aus alphanumerischen Zeichen, Unterstrichen, Bindestrichen und Punkten sein.

- Ereignisdetails
  - Enthält die eigentlichen Daten, die Sie in CloudWatch RUM aufzeichnen möchten.
  - Muss ein Objekt sein, das aus Feldern und Werten besteht.

Beispiele für die Aufzeichnung benutzerdefinierter Ereignisse

Es gibt zwei Möglichkeiten, benutzerdefinierte Ereignisse im CloudWatch RUM-Webclient aufzuzeichnen.

- Verwenden Sie die `recordEvent` API des CloudWatch RUM-Webclients.
- Verwenden Sie ein benutzerdefiniertes Plugin.

Ein benutzerdefiniertes Ereignis mithilfe der **`recordEvent`**-API senden, NPM-Beispiel

```
awsRum.recordEvent('my_custom_event', {
        location: 'IAD',
        current_url: 'amazonaws.com',
        user_interaction: {
            interaction_1 : "click",
            interaction_2 : "scroll"
        },
        visit_count:10
    }
)
```

Ein benutzerdefiniertes Ereignis mithilfe der **`recordEvent`**-API senden, Beispiel für eingebettetes Skript

```
cwr('recordEvent', {
    type: 'my_custom_event',
    data: {
        location: 'IAD',
        current_url: 'amazonaws.com',
        user_interaction: {
            interaction_1 : "click",
            interaction_2 : "scroll"
        },
        visit_count:10
    }
```

```
})
```

Beispiel für das Senden eines benutzerdefinierten Ereignisses mit einem benutzerdefinierten Plugin

```
// Example of a plugin that listens to a scroll event, and
// records a 'custom_scroll_event' that contains the timestamp of the event.
class MyCustomPlugin implements Plugin {
    // Initialize MyCustomPlugin.
    constructor() {
        this.enabled;
        this.context;
        this.id = 'custom_event_plugin';
    }
    // Load MyCustomPlugin.
    load(context) {
        this.context = context;
        this.enable();
    }
    // Turn on MyCustomPlugin.
    enable() {
        this.enabled = true;
        this.addEventHandler();
    }
    // Turn off MyCustomPlugin.
    disable() {
        this.enabled = false;
        this.removeEventHandler();
    }
    // Return MyCustomPlugin Id.
    getPluginId() {
        return this.id;
    }
    // Record custom event.
    record(data) {
        this.context.record('custom_scroll_event', data);
    }
    // EventHandler.
    private eventHandler = (scrollEvent: Event) => {
        this.record({timestamp: Date.now()})
    }
    // Attach an eventHandler to scroll event.
    private addEventHandler(): void {
        window.addEventListener('scroll', this.eventHandler);
```

```
    }
    // Detach eventHandler from scroll event.
    private removeEventHandler(): void {
        window.removeEventListender('scroll', this.eventHandler);
    }
}
```

Das CloudWatch RUM-Dashboard anzeigen

CloudWatch RUM hilft Ihnen dabei, Daten aus Benutzersitzungen über die Leistung Ihrer Anwendung zu sammeln, darunter Seitenladezeiten, Apdex-Score, verwendete Browser und Geräte, Geolokalisierung von Benutzersitzungen und fehlerhafte Sitzungen. All diese Informationen werden in einem Dashboard angezeigt.

RUM-Dashboard anzeigen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, RUM aus.

   Die Registerkarte Overview (Übersicht) zeigt Informationen an, die von einer von Ihnen erstellten App-Überwachungen gesammelt wurden.

   In der oberen Reihe werden die folgenden Informationen für diese App-Überwachung angezeigt:

   - Anzahl der Seitenladungen
   - Durchschnittliche Seitenladegeschwindigkeit
   - Apdex-Punktzahl
   - Status aller Alarme, die mit der App-Überwachung verknüpft sind

   Der Wert des Application Performance Index (Apdex) gibt die Zufriedenheit der Endbenutzer an. Die Punktzahl reicht von 0 (am wenigsten zufrieden) bis 1 (am zufriedensten). Die Ergebnisse basieren nur auf der Anwendungsleistung. Benutzer werden nicht aufgefordert, die Anwendung zu bewerten. Weitere Informationen über Apdex-Ergebnisse finden Sie unter Wie CloudWatch RUM die Apdex-Werte festlegt.

   Mehrere dieser Bereiche enthalten Links, mit denen Sie die Daten weiter untersuchen können. Wenn Sie einen dieser Links auswählen, wird eine detaillierte Ansicht mit den Registerkarten

Leistung, Fehler, HTTP-Anfragen, Sitzungen, Ereignisse, Browser & Geräte und User Journey oben in der Anzeige angezeigt.

3. Wählen Sie zur weiteren Eingrenzung die Registerkarte List view (Listenansicht) und danach den Namen der App-Überwachung aus, auf die Sie sich konzentrieren möchten. Dadurch werden die folgenden Registerkarten für die ausgewählte App-Überwachung angezeigt:

   - Die Registerkarte Performance (Leistung) zeigt Informationen zur Seitenleistung an, einschließlich Ladezeiten, Sitzungsinformationen, Anforderungsinformationen, Webdaten und Seitenladezeiten im Zeitverlauf. Diese Ansicht enthält Steuerelemente zum Umschalten der Ansicht zwischen Page loads (Seitenladungen), Requests (Anforderungen) und Location (Standort).
   - Auf der Registerkarte Fehler werden Informationen zu Javascript-Fehlern angezeigt, einschließlich der von Benutzern am häufigsten gesehenen Fehlermeldung sowie der Geräte und Browser mit den meisten Fehlern. Diese Ansicht umfasst ein Histogramm der Fehler und eine Listenansicht der Fehler. Sie können die Fehlerliste nach Benutzer- und Ereignisdetails filtern. Wählen Sie eine Fehlermeldung aus, um weitere Details zu sehen.
   - Auf der Registerkarte HTTP-Anfragen werden Informationen zu HTTP-Anfragen angezeigt, einschließlich der Anfrage-URL mit den meisten Fehlern und der Geräte und Browser mit den meisten Fehlern. Diese Registerkarte enthält ein Histogramm der Anfragen, eine Listenansicht der Anfragen und eine Listenansicht der Netzwerkfehler. Sie können die Listen nach Benutzer- und Ereignisdetails filtern. Wählen Sie einen Antwortcode oder eine Fehlermeldung, um weitere Details über die Anfrage bzw. den Netzwerkfehler anzuzeigen.
   - Auf der Registerkarte Sitzungen werden Sitzungsmetriken angezeigt. Diese Registerkarte enthält ein Histogramm der Startereignisse von Sitzungen und eine Listenansicht der Sitzungen. Sie können die Sitzungsliste nach Ereignistyp, Benutzerdetails und Ereignisdetails filtern. Wählen Sie eine sessionId, um weitere Details zu einer Sitzung zu sehen.
   - Auf der Registerkarte Ereignisse werden ein Histogramm der RUM-Ereignisse und eine Listenansicht der Ereignisse angezeigt. Sie können die Liste der Ereignisse nach Ereignistyp, Benutzerdetails und Ereignisdetails filtern. Wählen Sie ein RUM-Ereignis aus, um das Rohereignis zu sehen.
   - Die Registerkarte Browsers & Devices zeigt Informationen wie Leistung und Verwendung verschiedener Browser und Geräte für den Zugriff auf Ihre Anwendung an. Diese Ansicht enthält Steuerelemente zum Umschalten der Ansicht zwischen Browsers und Geräte.

     Wenn Sie den Umfang auf einen einzelnen Browser einschränken, werden die Daten nach Browserversion aufgeschlüsselt.

- Die User Journey (Benutzer-Journey) zeigt die Pfade an, die Ihre Kunden verwenden, um in Ihrer Anwendung zu navigieren. Sie können sehen, auf welcher Seite Kunden in Ihrer Anwendung starten und auf welcher Seite sie die Anwendung wieder verlassen. Sie können auch die Pfade sehen, die sie einschlagen, und den Prozentsatz der Kunden, die diesen Pfaden folgen. Sie können auf einem Knoten pausieren, um weitere Details zu dieser Seite zu erhalten. Sie können einen einzelnen Pfad auswählen, um die Verbindungen zur einfacheren Anzeige hervorzuheben.

4. (Optional) Auf einer der ersten sechs Registerkarten können Sie die Schaltfläche Seiten und anschließend eine Seite oder Seitengruppe aus der Liste auswählen. Dadurch werden die angezeigten Daten auf eine einzelne Seite oder auf eine Gruppe von Seiten Ihrer Anwendung beschränkt. Sie können auch Seiten und Seitengruppen in der Liste als Favoriten markieren.

Wie CloudWatch RUM die Apdex-Werte festlegt

Apdex (Application Performance Index) ist ein offener Standard, der eine Methode zum Melden, Benchmarking und Bewerten der Reaktionszeit von Anwendungen definiert. Ein Apdex-Score hilft Ihnen, die Auswirkungen auf die Anwendungsleistung im Laufe der Zeit zu verstehen und zu identifizieren.

Der Apdex-Score gibt an, dass der Zufriedenheitsgrad der Endbenutzer von 0 (am wenigsten zufrieden) bis 1 (am zufriedensten) reicht. Die Ergebnisse basieren nur auf der Anwendungsleistung. Benutzer werden nicht aufgefordert, die Anwendung zu bewerten.

Jeder einzelne Apdex-Score gehört zu einem von drei Schwellenwerten. Basierend auf dem Apdex-Schwellenwert und der tatsächlichen Reaktionszeit der Anwendung gibt es die folgenden drei Arten von Leistung:

- Zufrieden – Die tatsächliche Reaktionszeit der Anwendung ist kleiner oder gleich dem Apdex-Schwellenwert. Für CloudWatch RUM liegt dieser Schwellenwert bei 2000 ms oder weniger.
- Tolerabel – Die tatsächliche Reaktionszeit der Anwendung ist größer als der Apdex-Schwellenwert, aber kleiner oder gleich dem Vierfachen des Apdex-Schwellenwerts. Für CloudWatch RUM liegt dieser Bereich zwischen 2000 und 8000 ms.
- Frustrierend – Die tatsächliche Reaktionszeit der Anwendung ist größer als das Vierfache des Apdex-Schwellenwerts. Für CloudWatch RUM liegt dieser Bereich bei über 8000 ms.

Der gesamte Apdex-Score von 0-1 wird nach folgender Formel berechnet:

```
(positive scores + tolerable scores/2)/total scores * 100
```

CloudWatch Metriken, die Sie mit CloudWatch RUM sammeln können

In der Tabelle in diesem Abschnitt sind die Metriken aufgeführt, die Sie automatisch mit CloudWatch RUM erfassen. Sie können diese Metriken in der CloudWatch Konsole sehen. Weitere Informationen finden Sie unter Anzeigen der verfügbaren Metriken.

Sie können optional auch erweiterte Messwerte an CloudWatch oder CloudWatch Eviently senden. Weitere Informationen finden Sie unter Erweiterte Metriken.

Diese Metriken werden im Metrik-Namespace namens `AWS/RUM` veröffentlicht. Alle der folgenden Metriken werden mit einer `application_name`-Dimension veröffentlicht. Der Wert dieser Dimension ist der Name der App-Überwachung. Einige Metriken werden auch mit zusätzlichen Dimensionen veröffentlicht, wie in der Tabelle aufgeführt.

Metrik	Einheit	Beschreibung
`HttpStatusCodeCount`	Anzahl	Die Anzahl der HTTP-Antworten in der Anwendung nach ihrem Antwortstatuscode. Zusätzliche Dimensionen: • `event_details.response.status` ist der Antwortstatuscode wie 200, 400, 404 usw. • `event_type` Der Ereignistyp. Derzeit ist der einzig mögliche

Metrik	Einheit	Beschreibung
		Wert für diese Dimension `http`.
`Http4xxCount`	Anzahl	Die Anzahl der HTTP-Antworten in der Anwendung mit dem Antwortstatuscode 4xx. Diese werden auf der Grundlage von `http_event` RUM-Ereignissen berechnet, die zu 4xx-Codes führen.
`Http5xxCount`	Anzahl	Die Anzahl der HTTP-Antworten in der Anwendung mit dem Antwortstatuscode 5xx. Diese werden auf der Grundlage von `http_event` RUM-Ereignissen berechnet, die zu 5xx-Codes führen.
`JsErrorCount`	Anzahl	Die Anzahl der aufgenommenen JavaScript Fehlererereignisse.

Metrik	Einheit	Beschreibung
`NavigationFrustratedCount`	Anzahl	Die Anzahl der Navigationsereignisse mit einer `duration` über dem Frustrationsschwellwert, der 8000 ms beträgt. Die Dauer von Navigationsereignissen wird in der `PerformanceNavigationDuration` - Metrik verfolgt.
`NavigationSatisfiedCount`	Anzahl	Die Anzahl der Navigationsereignisse mit einer `duration`, die geringer ist als das Apdex-Ziel, das 2000 ms beträgt. Die Dauer von Navigationsereignissen wird in der `PerformanceNavigationDuration` - Metrik verfolgt.

Metrik	Einheit	Beschreibung
`NavigationToleratedCount`	Anzahl	Die Anzahl der Navigationsereignisse mit einer `duration` zwischen 2000 ms und 8000 ms. Die Dauer von Navigationsereignissen wird in der `PerformanceNavigationDuration` -Metrik verfolgt.
`PageViewCount`	Anzahl	Die Anzahl der Seitenaufrufereignisse, die vom App-Monitor aufgenommen wurden. Dies wird berechnet, indem die `page_view_event` RUM-Ereignisse gezählt werden.

Metrik	Einheit	Beschreibung
`PerformanceResourceDuration`	Millisekunden	Die `duration` eines Ressourcenereignisses. Zusätzliche Dimensionen: • `event_details.file.type` ist der Dateityp des Ressourcenereignisses, z. B. eine Stilvorlage, ein Dokument, ein Image, ein Skript oder eine Schriftart. • `event_type` Der Ereignistyp. Derzeit ist der einzig mögliche Wert für diese Dimension `resource`.
`PerformanceNavigationDuration`	Millisekunden	Die `duration` eines Navigationsereignisses.

Metrik	Einheit	Beschreibung
`RumEventPayloadSize`	Bytes	Die Größe jedes von CloudWatch RUM aufgenommenen Ereignisses. Sie können auch die `SampleCount` -Statistik für diese Metrik verwenden, um die Anzahl der Ereignisse zu überwachen, die eine App-Überwachung erfasst.
`SessionCount`	Anzahl	Die Anzahl der vom App-Monitor aufgenommenen Sitzungsstart-Ereignisse. Mit anderen Worten: Die Anzahl der neu gestarteten Sitzungen.
`WebVitalsCumulativeLayoutShift`	None	Verfolgt den Wert der kumulativen Layoutverschiebungs-Ereignisse.
`WebVitalsFirstInputDelay`	Millisekunden	Verfolgt den Wert der ersten Eingabeverzögerungs-Ereignisse.

Metrik	Einheit	Beschreibung
`WebVitalsLargestContentfulPaint`	Millisekunden	Verfolgt den Wert der größten inhaltlichen Paint-Ereignisse.

Benutzerdefinierte Metriken und erweiterte Metriken, die Sie an CloudWatch und CloudWatch Evidenly senden können

Standardmäßig senden RUM-App-Monitore Metriken an. CloudWatch Diese Standardmetriken und -dimensionen sind in CloudWatch Metriken aufgeführt, die Sie mit CloudWatch RUM sammeln können.

Sie können auch einen App-Monitor für den Export von Metriken einrichten. Der App Monitor kann erweiterte Metriken, benutzerdefinierte Metriken oder beides senden. Es kann sie an CloudWatch oder an CloudWatch Evidenly oder an beide senden.

- Benutzerdefinierte Metriken – Benutzerdefinierte Metriken sind Metriken, die Sie definieren. Bei benutzerdefinierten Metriken können Sie einen beliebigen Metriknamen und Namespace verwenden. Um die Metriken zu erhalten, können Sie beliebige benutzerdefinierte Ereignisse, integrierte Ereignisse, benutzerdefinierte Attribute oder Standardattribute verwenden.

  Sie können benutzerdefinierte Messwerte sowohl an beide als auch an CloudWatch CloudWatch Evidently senden.
- Erweiterte Metriken — Ermöglicht es Ihnen, die CloudWatch Standard-RUM-Metriken an Eviently zu senden, damit sie CloudWatch in Evidenly-Experimenten verwendet werden können. Sie können auch jede der CloudWatch Standard-RUM-Metriken CloudWatch mit zusätzlichen Dimensionen an senden. Auf diese Weise können diese Metriken Ihnen einen detaillierteren Überblick verschaffen.

Themen

- Benutzerdefinierte Metriken
- Erweiterte Metriken

Benutzerdefinierte Metriken

Um benutzerdefinierte Metriken zu senden, müssen Sie die AWS APIs oder AWS CLI anstelle der Konsole verwenden. Weitere Informationen zur Verwendung der AWS APIs finden Sie unter PutRumMetricsDestinationund BatchCreateRumMetricDefinitions.

Die maximale Anzahl von erweiterten und benutzerdefinierten Metrikdefinitionen, die ein Ziel enthalten kann, beträgt 2 000. Für jede benutzerdefinierte Metrik oder erweiterte Metrik, die Sie an jedes Ziel senden, zählt jede Kombination aus Dimensionsname und Dimensionswert für dieses Limit. Dies gilt auch als CloudWatch benutzerdefinierte Kennzahl für die Preisgestaltung.

Das folgende Beispiel zeigt, wie Sie eine benutzerdefinierte Metrik erstellen, die von einem benutzerdefinierten Ereignis abgeleitet ist. Hier das Beispiel für das verwendete benutzerdefinierte Ereignis:

```
cwr('recordEvent', {
    type: 'my_custom_event',
    data: {
        location: 'IAD',
        current_url: 'amazonaws.com',
        user_interaction: {
            interaction_1 : "click",
            interaction_2 : "scroll"
        },
        visit_count:10
    }
})
```

Mit diesem benutzerdefinierten Ereignis können Sie eine benutzerdefinierte Metrik erstellen, die die Anzahl der Besuche auf der URL `amazonaws.com` von Chrome-Browsern aus zählt. Die folgende Definition erstellt eine Metrik namens `AmazonVisitsCount` in Ihrem Konto im `RUM/CustomMetrics/PageVisits`-Namespace.

```
{
    "AppMonitorName":"customer-appMonitor-name",
    "Destination":"CloudWatch",
    "MetricDefinitions":[
        {
            "Name":"AmazonVisitsCount",
            "Namespace":"PageVisit",
            "ValueKey":"event_details.visit_count",
```

```
            "UnitLabel":"Count",
            "DimensionKeys":{
                "event_details.current_url": "URL"
            },
            "EventPattern":"{\"metadata\":{\"browserName\":[\"Chrome\"]},\"event_type
\":[\"my_custom_event\"],\"event_details\": {\"current_url\": [\"amazonaws.com\"]}}"
        }
    ]
}
```

Erweiterte Metriken

Wenn Sie erweiterte Metriken einrichten, können Sie eine oder beide der folgenden Aktionen ausführen:

- Senden Sie CloudWatch Standard-RUM-Metriken an CloudWatch Eviently, um sie in Evidenly-Experimenten zu verwenden. Nur die WebVitalsLargestContentfulPaintMetriken PerformanceNavigationDurationPerformanceResourceDuration, WebVitalsCumulativeLayoutShift, WebVitalsFirstInputDelay, und können an Evidently gesendet werden.
- Senden Sie eine beliebige der CloudWatch RUM-Standardmetriken CloudWatch mit zusätzlichen Dimensionen an, sodass Sie anhand der Metriken einen detaillierteren Überblick erhalten. Sie können beispielsweise Metriken sehen, die für einen bestimmten Browser spezifisch sind, der von Ihren Benutzern verwendet wird, oder Metriken für Benutzer an einem bestimmten Standort.

Weitere Informationen zu den CloudWatch Standard-RUM-Metriken finden Sie unter. CloudWatch Metriken, die Sie mit CloudWatch RUM sammeln können

Die maximale Anzahl von erweiterten und benutzerdefinierten Metrikdefinitionen, die ein Ziel enthalten kann, beträgt 2 000. Für jede erweiterte oder benutzerdefinierte Metrik, die Sie an jedes Ziel senden, zählt jede Kombination aus Dimensionsname und Dimensionswert als eine erweiterte Metrik für dieses Limit. Dies gilt auch als CloudWatch benutzerdefinierte Kennzahl für die Preisgestaltung.

Wenn Sie erweiterte Messwerte an senden CloudWatch, können Sie die CloudWatch RUM-Konsole verwenden, um CloudWatch Alarme für sie zu erstellen.

Erweiterte Metriken werden als CloudWatch benutzerdefinierte Metriken berechnet. Weitere Informationen finden Sie unter Amazon CloudWatch -Preisgestaltung.

Die folgenden Dimensionen werden für erweiterte Metriken für alle Metriknamen unterstützt, die Anwendungsmonitoren senden können. Diese Metriknamen sind in CloudWatch Metriken, die Sie mit CloudWatch RUM sammeln können aufgeführt.

- `BrowserName`

  Beispiel für Dimensionswerte: `Chrome`, `Firefox`, `Chrome Headless`
- `CountryCode` Dabei wird das ISO-3166-Format mit zweibuchstabigen Codes verwendet.

  Beispiel für Dimensionswerte: `US`, `JP`, `DE`
- `DeviceType`

  Beispiel für Dimensionswerte: `desktop`, `mobile`, `tablet`, `embedded`
- `FileType`

  Beispiel für Dimensionswerte: `Image`, `Stylesheet`
- `OSName`

  Beispiel für Dimensionswerte: `Linux`, `Windows, `, `iOS`, `Android`
- `PageId`

Erweiterte Metriken über die Konsole einrichten

Gehen Sie wie folgt vor, um die Konsole zum Senden erweiterter Messwerte zu CloudWatch verwenden.

Um erweiterte Messwerte an CloudWatch Evidently zu senden, müssen Sie die AWS APIs oder AWS CLI anstelle der Konsole verwenden. Informationen zur Verwendung der AWS APIs zum Senden erweiterter Messwerte an entweder CloudWatch oder Eviently finden Sie unter PutRumMetricsDestinationund. BatchCreateRumMetricDefinitions

So richten Sie mit der Konsole einen App-Monitor ein und senden erweiterte RUM-Metriken an CloudWatch

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, RUM aus.
3. Wählen Sie List view (Listenansicht) und danach den Namen des Anwendungsmonitors aus, die Metriken senden soll.

4. Wählen Sie den Tab Configuration (Konfiguration) und wählen Sie dann RUM extended metrics (erweiterte RUM-Metriken) aus.
5. Wählen Sie Send metrics (Metriken senden) aus.
6. Wählen Sie einen oder mehrere Metriknamen aus, um sie mit zusätzlichen Dimensionen zu senden.
7. Wählen Sie einen oder mehrere Faktoren aus, die als Dimensionen für diese Metriken verwendet werden sollen. Während Sie Ihre Auswahl treffen, wird die Anzahl der erweiterten Metriken, die durch Ihre Auswahl erstellt wurden, unter Number of extended metrics (Anzahl der erweiterten Metriken) angezeigt.

   Diese Zahl wird berechnet, indem die Anzahl der ausgewählten Metriknamen mit der Anzahl der verschiedenen Dimensionen, die Sie erstellen, multipliziert wird. Diese Zahl gibt an, wie viele benutzerdefinierte Metriken Ihnen in Rechnung gestellt werden. Weitere Informationen zur CloudWatch Preisgestaltung finden Sie unter CloudWatchAmazon-Preise.

   a. Um eine Metrik mit Seiten-ID als Dimension zu senden, wählen Sie Browse for page ID (Nach Seiten-ID suchen) und wählen Sie dann die zu verwendenden Seiten-IDs aus.
   b. Um eine Metrik mit dem Gerätetyp als Dimension zu senden, wählen Sie entweder Desktop devices (Desktop-Geräte) oder Mobile and tablets (Mobil und Tablets) aus.
   c. Um eine Metrik mit dem Betriebssystem als Dimension zu senden, wählen Sie unter Operating system (Betriebssystem) ein oder mehrere Betriebssysteme aus.
   d. Um eine Metrik mit dem Browsertyp als Dimension zu senden, wählen Sie unter Browser einen oder mehrere Browsers (Browser) aus.
   e. Um eine Metrik mit Geolokalisierung als Dimension zu senden, wählen Sie unter Standorte einen oder mehrere Locations (Standorte) aus.

      Nur die Standorte, von denen dieser Anwendungsmonitor Messwerte gemeldet hat, werden in der Liste angezeigt, aus denen Sie auswählen können.
8. Wenn Sie mit Ihrer Auswahl fertig sind, wählen Sie Send metrics (Metriken senden) aus.
9. (Optional) Um einen Alarm zu erstellen, der eine der Metriken überwacht, wählen Sie in der Liste der Extended metrics (erweiterte Metriken) in der Zeile dieser Metrik die Option Create alarm (Alarm erstellen) aus.

   Allgemeine Informationen zu CloudWatch Alarmen finden Sie unter CloudWatch Amazon-Alarme verwenden. Eine Anleitung zum Einrichten eines Alarms für eine erweiterte CloudWatch RUM-Metrik finden Sie unterTutorial: Eine erweiterte Metrik erstellen und sie alarmieren.

Das Senden erweiterter Metriken beenden

So verwenden Sie die Konsole, um das Senden erweiterter Metriken zu beenden

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, RUM aus.
3. Wählen Sie List view (Listenansicht) und danach den Namen des Anwendungsmonitors aus, die Metriken senden soll.
4. Wählen Sie den Tab Configuration (Konfiguration) und wählen Sie dann RUM extended metrics (erweiterte RUM-Metriken) aus.
5. Wählen Sie eine oder mehrere Kombinationen aus Metriknamen und Dimensionen aus, um das Senden zu beenden. Wählen Sie dann Actions (Aktionen), Delete (Löschen).

Tutorial: Eine erweiterte Metrik erstellen und sie alarmieren

In diesem Tutorial wird gezeigt, wie Sie eine erweiterte Metrik einrichten, an die gesendet werden soll CloudWatch, und anschließend, wie Sie für diese Metrik einen Alarm einrichten. In diesem Tutorial erstellen Sie eine Metrik, die JavaScript Fehler im Chrome-Browser verfolgt.

So richten Sie diese erweiterte Metrik ein und stellen einen Alarm dafür ein

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, RUM aus.
3. Wählen Sie List view (Listenansicht) und danach den Namen des Anwendungsmonitors aus, die Metriken senden soll.
4. Wählen Sie den Tab Configuration (Konfiguration) und wählen Sie dann RUM extended metrics (erweiterte RUM-Metriken) aus.
5. Wählen Sie Send metrics (Metriken senden) aus.
6. Wählen Sie JS ausErrorCount.
7. Wählen Sie unter Browsers (Browser) Chrome (Chrome) aus.

   Diese Kombination aus JS ErrorCount und Chrome sendet eine erweiterte Metrik an CloudWatch. Die Metrik zählt JavaScript Fehler nur für Benutzersitzungen, die den Chrome-Browser verwenden. Der Metrikname JsErrorCountund der Name der Dimension lauten Browser.
8. Wählen Sie Send metrics (Metriken senden) aus.

9. Wählen Sie in der Liste der erweiterten Messwerte in der Zeile, die JsErrorCountunter Name und Chrome unter angezeigt wird, die Option Alarm erstellen aus BrowserName.
10. Vergewissern Sie sich unter Metrik und Bedingungen angeben, dass der Metrikname und die BrowserNameFelder vorab mit den richtigen Werten gefüllt sind.
11. Wählen Sie unter Statistic (Statistik) die Statistik aus, die Sie für den Alarm verwenden möchten. Der Average (Durchschnitt) ist eine gute Wahl für diese Art von Zählmetrik.
12. Wählen Sie unter Zeitraum die Option 5 Minuten aus.
13. Führen Sie unter Bedingungen die folgenden Schritte aus:

    - Wählen Sie Static (Statisch).
    - Wählen Sie Greater (Größer), um festzulegen, dass der Alarm in den Zustand ALARM wechseln soll, wenn die Anzahl der Fehler den Schwellenwert überschreitet, den Sie gerade angeben möchten.
    - Unter than ... (als ...), geben Sie die Zahl für den Alarmschwellenwert ein. Der Alarm geht in den Zustand ALARM über, wenn die Anzahl der Fehler über einen Zeitraum von 5 Minuten diesen Wert überschreitet.
14. (Optional) Standardmäßig wechselt der Alarm in den Zustand ALARM, sobald die Anzahl der Fehler den von Ihnen festgelegten Schwellenwert innerhalb eines Zeitraums von 5 Minuten überschreitet. Sie können dies optional ändern, sodass der Alarm nur dann in den Zustand ALARM wechselt, wenn diese Zahl für mehr als einen Zeitraum von 5 Minuten überschritten wird.

    Wählen Sie dazu Additional configuration (Zusätzliche Konfiguration) und geben Sie dann für Datapoints to alarm (zu alarmierende Datenpunkte) an, wie viele 5-Minuten-Zeiträume die Fehlernummer über dem Schwellenwert liegen muss, um den Alarm auszulösen. Sie können beispielsweise 2 von 2 auswählen, damit der Alarm nur ausgelöst wird, wenn zwei aufeinanderfolgende 5-Minuten-Zeiträume den Schwellenwert überschreiten, oder 2 von 3, um den Alarm auszulösen, wenn zwei von drei aufeinanderfolgenden 5-Minuten-Zeiträumen den Schwellenwert überschreiten.

    Weitere Informationen zu den verschiedenen Arten der Alarmauswertung finden Sie unter Auswerten eines Alarms.
15. Wählen Sie Weiter aus.
16. Geben Sie unter Configure actions (Aktionen konfigurieren) an, was passieren soll, wenn der Alarm in den Alarmzustand wechselt. Führen Sie Folgendes aus, um eine Benachrichtigung mit Amazon SNS zu erhalten:

- Wählen Sie Add notification (Benachrichtigung hinzufügen) aus.
- Wählen Sie In Alarm aus.
- Wählen Sie ein vorhandenes SNS-Thema aus oder erstellen Sie ein neues Thema. Wenn Sie eine neue erstellen, geben Sie einen Namen für sie an und fügen Sie mindestens eine E-Mail-Adresse hinzu.

17. Wählen Sie Weiter aus.
18. Geben Sie einen Namen und optional eine Beschreibung für den Alarm ein und wählen Sie dann Next (Weiter).
19. Prüfen Sie die Details und wählen Sie Create alarm (Alarm erstellen).

Datenschutz und Datenschutz bei RUM CloudWatch

Das Modell der AWS gemeinsamen Verantwortung gilt für den Datenschutz und den Datenschutz in Amazon CloudWatch RUM. AWS Ist, wie in diesem Modell beschrieben, für den Schutz der globalen Infrastruktur verantwortlich, auf der die gesamte AWS Cloud läuft. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blogbeitrag The AWS Shared Responsibility Model und GDPR im AWS Security Blog. Weitere Ressourcen zur Einhaltung der DSGVO-Anforderungen finden Sie im Center für Datenschutz-Grundverordnung (DSGVO).

Amazon CloudWatch RUM generiert auf der Grundlage Ihrer Eingabe von Endbenutzerdaten, die Sie sammeln möchten, einen Codeausschnitt, den Sie in Ihren Website- oder Webanwendungscode einbetten können. Der Webclient, der durch den Codeausschnitt heruntergeladen und konfiguriert wurde, verwendet Cookies (oder ähnliche Technologien), um Ihnen bei der Erfassung von Endbenutzerdaten zu helfen. Die Verwendung von Cookies (oder ähnlichen Technologien) unterliegt in bestimmten Gerichtsbarkeiten den Datenschutzbestimmungen. Bevor Sie Amazon CloudWatch RUM verwenden, empfehlen wir Ihnen dringend, Ihre Verpflichtungen zur Einhaltung der geltenden Gesetze zu überprüfen, einschließlich aller geltenden rechtlichen Anforderungen, um rechtlich angemessene Datenschutzhinweise bereitzustellen und alle erforderlichen Einwilligungen für die Verwendung von Cookies und die Verarbeitung (einschließlich Erfassung) von Endnutzerdaten einzuholen. Weitere Informationen darüber, wie der Webclient Cookies (oder ähnliche Technologien) verwendet und welche Endnutzerdaten der Webclient sammelt, finden Sie unter Vom RUM-Webclient gesammelte Informationen CloudWatch und. CloudWatch RUM-Webclient-Cookies (oder ähnliche Technologien)

Wir empfehlen dringend, in Freitextfeldern keine sensiblen personenbezogenen Daten wie Kontonummern von Endkunden, E-Mail-Adressen oder sonstige personenbezogenen Daten einzugeben. Alle Daten, die Sie in Amazon CloudWatch RUM oder andere Dienste eingeben, können in Diagnoseprotokollen enthalten sein.

CloudWatch RUM-Webclient-Cookies (oder ähnliche Technologien)

Der CloudWatch RUM-Webclient sammelt standardmäßig bestimmte Daten über Benutzersitzungen. Sie können Cookies aktivieren, damit der Webclient eine Benutzer-ID und eine Sitzungs-ID sammelt, die auch beim Laden der Seite bestehen bleiben. Die Benutzer-ID wird nach dem Zufallsprinzip von RUM generiert.

Wenn diese Cookies aktiviert sind, kann RUM die folgenden Datentypen anzeigen, wenn Sie das RUM-Dashboard für diese App-Überwachung anzeigen.

- Aggregierte Daten basierend auf Benutzer-IDs, wie die Anzahl der individuellen Benutzer und die Anzahl der verschiedenen Benutzer, bei denen ein Fehler aufgetreten ist.
- Aggregierte Daten basierend auf Sitzungs-IDs, wie die Anzahl der Sitzungen und die Anzahl der Sitzungen, bei denen ein Fehler aufgetreten ist.
- Die User Journey (Benutzer-Journey) bezieht sich auf die Reihenfolge der Seiten, die jede ausgewählte Benutzersitzung enthält.

> **Important**
>
> Wenn Sie diese Cookies (oder ähnliche Technologien) nicht aktivieren, zeichnet der Webclient dennoch bestimmte Informationen über Endbenutzersitzungen auf, z. B. Browsertyp/-version, Betriebssystemtyp/-version, Gerätetyp usw. Diese werden gesammelt, um aggregierte seitenspezifische Erkenntnisse wie Webdaten, Seitenaufrufe und Seiten mit Fehlern darzustellen. Weitere Informationen zu den aufgezeichneten Daten finden Sie unter Vom RUM-Webclient gesammelte Informationen CloudWatch .

Vom RUM-Webclient gesammelte Informationen CloudWatch

In diesem Abschnitt wird das PutRumEventsSchema dokumentiert, das die Struktur der Daten definiert, die Sie in Benutzersitzungen mit CloudWatch RUM sammeln können.

Eine PutRumEventsAnfrage sendet eine Datenstruktur mit den folgenden Feldern an CloudWatch RUM.

- Die ID dieser RUM-Ereignisse
- Details zur App-Überwachung mit folgendem Inhalt:
  - App-Überwachungs-ID
  - Überwachte Anwendungsversion
- Details zur Benutzern mit folgendem Inhalt: Dies wird nur erfasst, wenn bei der App-Überwachung Cookies aktiviert sind.
  - Eine vom Webclient generierte Benutzer-ID
  - Sitzungs-ID
- Das Array von RUM-Ereignissen in diesem Batch.

RUM-Ereignisschema

Die Struktur jedes RUM-Ereignisses umfasst die folgenden Felder.

- Die Ereignis-ID
- Einen Zeitstempel
- Einen Ereignistyp
- Den Benutzer-Agent
- Metadaten
- Details zu RUM-Ereignissen

RUM-Ereignismetadaten

Die Metadaten umfassen Seitenmetadaten, Metadaten des Benutzeragenten, Geolokalisierungs-Metadaten und Domain-Metadaten.

Seiten-Metadaten

Die Seiten-Metadaten enthalten Folgendes:

- Seiten-ID
- Seitentitel

- ID der übergeordneten Seite. – Dies wird nur erfasst, wenn bei der App-Überwachung Cookies aktiviert sind.
- Interaktionstiefe – Dies wird nur erfasst, wenn bei der App-Überwachung Cookies aktiviert sind.
- Seiten-Tags – Sie können den Seitenereignissen Tags hinzufügen, um Seiten zu gruppieren. Weitere Informationen finden Sie unter Verwenden von Seitengruppen.

Benutzer-Agent-Metadaten

Die Benutzer-Metadaten enthalten Folgendes:

- Browser-Sprache
- Browsername
- Browserversion
- Name des Betriebssystems
- Version des Betriebssystems
- Gerätetyp
- Plattformtype

Geolocation-Metadaten

Die Geolocation-Metadaten enthalten Folgendes:

- Ländercode
- Unterteilungscode

Domain-Metadaten

Die Domain-Metadaten enthalten die URL-Domain.

Details zu RUM-Ereignissen

Die Details eines Ereignisses folgen je nach Ereignistyp einem der folgenden Arten von Schemas.

Sitzungsstart-Ereignis

Dieses Ereignis enthält keine Felder. Dies wird nur erfasst, wenn bei der App-Überwachung Cookies aktiviert sind.

Seitenansicht-Schema

Ein Seitenansicht-Ereignis enthält die folgenden Eigenschaften: Sie können die Sammlung von Seitenansichten deaktivieren, indem Sie den Webclient konfigurieren. Weitere Informationen finden Sie in der CloudWatch RUM-Webclient-Dokumentation.

Name	Typ	Beschreibung
Seiten-ID	String	Eine ID, die diese Seite innerhalb der Anwendung eindeutig darstellt. Standardmäßig ist dies der URL-Pfad.
ID der übergeordneten Seite	String	Die ID der Seite, auf der sich der Benutzer befand, als er zur aktuellen Seite navigierte. Dies wird nur erfasst, wenn bei der App-Überwachung Cookies aktiviert sind.
Interaktionstiefe	String	Dies wird nur erfasst, wenn bei der App-Überwachung Cookies aktiviert sind.

JavaScript Fehlerschema

JavaScript Vom Agenten generierte Fehlerereignisse enthalten die folgenden Eigenschaften. Der Webclient sammelt diese Ereignisse nur, wenn Sie sich entschieden haben, die Fehlertelemetrie zu sammeln.

Name	Typ	Beschreibung
Fehlertyp	String	Der Name des Fehlers, sofern vorhanden. Weitere Informationen finden Sie unter error.prototype.name. Einige Browser unterstützen möglicherweise keine Fehlertypen.
Fehlermeldung	String	Die Fehlermeldung. Weitere Informationen finden Sie unter Error.prototype.message. Wenn das Fehlerfeld nicht vorhanden ist, ist dies die Meldung des Fehlerereignisses. Weitere Informationen finden Sie unter ErrorEvent.

Name	Typ	Beschreibung
		Fehlermeldungen sind möglicherweise in verschiedenen Browsern nicht konsistent.
Stack-Abl aufverfolgung	String	Die Stack-Ablaufverfolgung des Fehlers wurde, falls vorhanden, auf 150 Zeichen gekürzt. Weitere Informati onen finden Sie unter Error.prototype.Stack. Einige Browser unterstützen möglicherweise keine Stack-Ablaufverfolgungen.

DOM-Ereignisschema

Vom Agent generierte Ereignisse des Document Object Model (DOM) enthalten die folgenden Eigenschaften. Diese Ereignisse werden standardmäßig nicht erfasst. Sie werden nur gesammelt, wenn Sie die Interaktionstelemetrie aktivieren. Weitere Informationen finden Sie in der CloudWatch RUM-Webclient-Dokumentation.

Name	Typ	Beschreibung
Veranstaltung	String	Der Typ des DOM-Ereignisses wie Klicken, Scrollen oder Hover. Weitere Informationen finden Sie unter Ereignisr eferenz.
Element	String	Der DOM-Elementtyp
Element-ID	String	Wenn das Element, das das Ereignis generiert hat, eine ID besitzt, speichert diese Eigenschaft diese ID. Weitere Informationen finden Sie unter Element.id.
CSSLocator	String	Der CSS-Locator, der zur Identifizierung des DOM-Eleme nts verwendet wird.
InteractionId	String	Eine eindeutige ID für die Interaktion zwischen dem Benutzer und der Benutzeroberfläche.

Navigationsereignisschema

Navigationsereignisse werden nur erfasst, wenn die App-Überwachung die Leistungstelemetrie aktiviert hat.

Navigationsereignisse verwenden APIs mit Navigationszeitpunkt Stufe 1 und Navigationszeitpunkt Stufe 2. APIs der Stufe 2 werden nicht in allen Browsern unterstützt, daher sind diese neueren Felder optional.

> **Note**
>
> Timestamp-Metriken basieren auf DOM HighResTimestamp. Bei APIs der Stufe 2 sind alle Timings standardmäßig relativ zum `startTime`. Aber für Stufe 1 wird die `navigationStart`-Metrik von Zeitstempelmetriken subtrahiert, um relative Werte zu erhalten. Alle Zeitstempelwerte sind in Millisekunden.

Navigationsereignisse enthalten die folgenden Eigenschaften.

Name	Typ	Beschreibung	Hinweise
initiatorType	String	Repräsentiert den Ressourcentyp, der das Leistungsereignis ausgelöst hat.	Wert: „Navigation“ Stufe 1 „Navigation“ Stufe 2 entryData .initiatorType
navigationType	String	Repräsentiert den Typ der Navigation. Dieses Attribut ist nicht erforderlich.	Wert: Dieser Wert muss einer der folgenden sein: • `navigate` ist eine

Name	Typ	Beschreibung	Hinweise
			Navigation, die durch die Auswahl eines Links, die Eingabe einer URL in die Adresszeile eines Browsers, das Senden eines Formulars oder durch das Initialisieren durch eine Skriptoperation, bei der es sich nicht um `reload` oder `back_forward` handelt, gestartet wird. • `reload` ist eine Navigation durch den

Name	Typ	Beschreibung	Hinweise
			Reload-Vorgang des Browsers oder durch `location.reload()`. • `back_forward` ist eine Navigation durch die Verlaufstraversierung des Browsers. • `prerender` ist eine Navigation, die durch einen Prerender-Hinweis initiiert wird. Weitere Informationen finden Sie unter Prerender.

Name	Typ	Beschreibung	Hinweise
startTime	Zahl	Gibt an, wann das Ereignis ausgelöst wird.	Wert: 0 Stufe 1: entryData.navigationStart – entryData.navigationStart Stufe 2: entryData.startTime

Name	Typ	Beschreibung	Hinweise
unloadEventStart	Zahl	Gibt den Zeitpunkt an, zu dem das vorherige Dokument im Fenster entladen wurde, nachdem das `unload`-Ereignis ausgegeben wurde.	Wert: Wenn kein vorheriges Dokument vorhanden ist oder wenn das vorherige Dokument oder eine der erforderlichen Weiterleitungen nicht denselben Ursprung haben, wird der Wert 0 zurückgegeben. Stufe 1: `entryData.unloadEventStart > 0 ? entryData.unloadEventStart - entryData.navigationStart : 0` Ebene 2: EntryData.

Name	Typ	Beschreibung	Hinweise
			unloadEvent Start

Name	Typ	Beschreibung	Hinweise
promptForUnload	Zahl	Die Zeit, die zum Entladen des Dokuments benötigt wird. Mit anderen Worten, die Zeit zwischen `unloadEventStart` und `unloadEventEnd` . `UnloadEventEnd` repräsentiert den Moment in Millisekunden, in dem der Entlade-Ereignishandler beendet ist.	Wert: Wenn kein vorheriges Dokument vorhanden ist oder wenn das vorherige Dokument oder eine der erforderlichen Weiterleitungen nicht denselben Ursprung haben, wird der Wert 0 zurückgegeben. Ebene 1: EntryData.unloadEventEnd - Eingabedaten. unloadEventStart Stufe 2: EntryData.unloadEventEnd - Eingabedaten.

Name	Typ	Beschreibung	Hinweise
			unloadEventStart
redirectCount	Zahl	Eine Zahl, die die Anzahl der Weiterleitungen seit der letzten Navigation ohne Weiterleitung im aktuellen Browser-Kontext darstellt. Dieses Attribut ist nicht erforderlich.	Wert: Wenn es keine Weiterleitung gibt oder wenn es eine Weiterleitung gibt, die nicht den gleichen Ursprung wie das Zieldokument hat, ist der zurückgegebene Wert 0. Stufe 1: Nicht verfügbar Stufe 2: entryData.redirectCount

Name	Typ	Beschreibung	Hinweise
redirectStart	Zahl	Der Zeitpunkt, zu dem die erste HTTP-Weiterleitung gestartet wird.	Wert: Wenn es keine Weiterleitung gibt oder wenn es eine Weiterleitung gibt, die nicht den gleichen Ursprung wie das Zieldokument hat, ist der zurückgegebene Wert 0. Stufe 1: `entryData.redirectStart > 0 ? entryData.redirectStart - entryData.navigationStart : 0` Stufe 2: entryData.redirectStart

Name	Typ	Beschreibung	Hinweise
redirectTime	Zahl	Die nötige Zeit für die HTTP-Umleitung. Dies ist die Differenz zwischen `redirectStart` und `redirectEnd` .	Stufe 1:: entryData .redirectEnd – entryData .redirectStart Stufe 2:: entryData .redirectEnd – entryData .redirectStart

Name	Typ	Beschreibung	Hinweise
workerStart	Zahl	Dies ist eine Eigenschaft der `PerformanceResourceTiming` -Schnittstelle. Dies markiert den Beginn der Worker-Thread-Operation. Dieses Attribut ist nicht erforderlich.	Wert: Wenn ein Service-Worker-Thread bereits ausgeführt wird oder unmittelbar vor dem Starten des Service-Worker-Threads, gibt diese Eigenschaft die Zeit unmittelbar vor dem Aussenden von `FetchEvent` zurück. Es gibt 0 zurück, wenn die Ressource nicht von einem Service Worker abgefangen wird. Stufe 1: Nicht verfügbar

Name	Typ	Beschreibung	Hinweise
			Stufe 2: entryData .workerStart
workerTime	Zahl	Wenn die Ressource von einem Service Worker abgefangen wird, gibt dies die Zeit zurück, die für den Worker-Thread-Betrieb benötigt wird. Dieses Attribut ist nicht erforderlich.	Stufe 1: Nicht verfügbar Stufe 2: `entryData .workerStart > 0 ? entryData .fetchStart - entryData .workerStart : 0`
fetchStart	Zahl	Der Zeitpunkt, zu dem der Browser bereit ist, das Dokument mit einer HTTP-Anforderung abzurufen. Dies geschieht, bevor Sie einen Anwendungscache überprüfen.	Stufe 1: `: entryData .fetchStart > 0 ? entryData .fetchStart - entryData .navigationStart : 0` Stufe 2: entryData .fetchStart

Name	Typ	Beschreibung	Hinweise
domainLookupStart	Zahl	Der Zeitpunkt, zu dem die Domainsuche beginnt.	Wert: Wenn eine dauerhafte Verbindung verwendet wird oder wenn die Informationen in einem Cache oder einer lokalen Ressource gespeichert sind, ist der Wert derselbe wie `fetchStart` . Stufe 1: `entryData.domainLookupStart > 0 ? entryData.domainLookupStart - entryData.navigationStart : 0` Stufe 2: EntryData.

Name	Typ	Beschreibung	Hinweise
			domainLookupStart
dns	Zahl	Die Zeit, die für die Domainsuche benötigt wird.	Wert: Wenn die Ressourcen und DNS-Datensätze zwischengespeichert werden, ist der erwartete Wert 0. Ebene 1: EntryData.domainLookupEnd - Eingabedaten.domainLookupStart Stufe 2: EntryData.domainLookupEnd - Eingabedaten.domainLookupStart

Name	Typ	Beschreibung	Hinweise
nextHopProtocol	String	Eine Zeichenfolge, die das zum Abrufen der Ressource verwendete Netzwerkprotokoll darstellt. Dieses Attribut ist nicht erforderlich.	Stufe 1: Nicht verfügbar Stufe 2: EntryData.nextHopProtocol

Name	Typ	Beschreibung	Hinweise
connectStart	Zahl	Die Zeit unmittelbar bevor der Benutzeragent mit dem Herstellen der Verbindung zum Server beginnt, um das Dokument abzurufen.	Wert: Wenn eine dauerhafte RFC2616-Verbindung verwendet wird oder wenn das aktuelle Dokument aus relevanten Anwendungscaches oder lokalen Ressourcen abgerufen wird, gibt dieses Attribut den Wert `domainLookupEnd` aus. Stufe 1: `entryData.connectStart > 0 ? entryData.connectStart - entryData.navigationStart`

Name	Typ	Beschreibung	Hinweise
			`: 0` Stufe 2: entryData .connectStart
connect	Zahl	Misst die Zeit, die zum Herstellen der Transport verbindungen oder zur Durchführung der SSL-Authentifizierung erforderlich ist. Es enthält auch die blockierte Zeit, die in Anspruch genommen wird, wenn zu viele gleichzeitige Anfragen vom Browser ausgegeben werden.	Stufe 1: entryData .connectEnd – entryData .connectStart Stufe 2: entryData .connectEnd – entryData .connectStart
secureCon nectionStart	Zahl	Wenn das URL-Schema der aktuellen Seite „https“ lautet, gibt dieses Attribut umgehend die Zeit zurück, bevor der Benutzeragent den Handshake-Prozess startet, um die aktuelle Verbindung zu sichern. Es gibt 0 zurück, wenn HTTPS nicht verwendet wird. Weitere Informati onen zu URL-Schemata finden Sie unter URL-Darstellung.	Formel: EntryData. secureCon nectionStart

Name	Typ	Beschreibung	Hinweise
tlsTime	Zahl	Die Zeit, die benötigt wird, um einen SSL-Handshake abzuschließen.	Stufe 1: `entryData.secureConnectionStart > 0 ? entryData.connectEnd - entryData.secureConnectionStart : 0` Stufe 2: `entryData.secureConnectionStart > 0 ? entryData.connectEnd - entryData.secureConnectionStart : 0`

Name	Typ	Beschreibung	Hinweise
requestStart	Zahl	Die Zeit unmittelbar bevor der Benutzeragent beginnt, die Ressource vom Server oder von relevanten Anwendungscaches oder von lokalen Ressourcen anzufordern.	Stufe 1: `: entryData.requestStart > 0 ? entryData.requestStart - entryData.navigationStart : 0` Stufe 2: entryData .requestStart
timeToFirstByte	Zahl	Die Zeit, die benötigt wird, um das erste Byte an Informationen zu erhalten, nachdem eine Anfrage gestellt wurde. Diese Zeit ist relativ zu `startTime` .	Stufe 1: entryData .responseStart – entryData .requestStart Stufe 2: entryData .responseStart – entryData .requestStart

Name	Typ	Beschreibung	Hinweise
responseStart	Zahl	Die Zeit unmittelbar nachdem der HTTP-Parser des Benutzeragenten das erste Byte der Antwort von den relevanten Anwendung scaches oder von lokalen Ressourcen oder vom Server erhalten hat.	Stufe 1: `entryData.responseStart > 0 ? entryData.responseStart - entryData.navigationStart : 0` Stufe 2: entryData.responseStart

Name	Typ	Beschreibung	Hinweise
responseTime	String	Die Zeit, die benötigt wird, um eine vollständige Antwort in Form von Bytes von den relevanten Anwendungs-Caches, von lokalen Ressourcen oder vom Server zu erhalten.	Stufe 1: `entryData.responseStart > 0 ? entryData.responseEnd - entryData.responseStart : 0` Stufe 2: `entryData.responseStart > 0 ? entryData.responseEnd - entryData.responseStart : 0`

Name	Typ	Beschreibung	Hinweise
domIntera ctive	Zahl	Der Zeitpunkt, zu dem der Parser seine Arbeit am Hauptdokument beendet hat und das HTML-DOM erstellt wurde. Zu diesem Zeitpunkt wechselt `Document. readyState` zu „interaktiv“ und das entsprechende `readystatechange` -Ereignis wird ausgegeben.	Stufe 1: `entryData .domInter active > 0 ? entryData .domInter active - entryData .navigati onStart : 0` Stufe 2: entryData .domInter active

Name	Typ	Beschreibung	Hinweise
domContentLoadedEventStart	Zahl	Stellt den Zeitwert dar, der der Zeit unmittelbar entspricht, bevor der Benutzeragent das ContentLoaded DOM-Ereignis im aktuellen Dokument auslöst. Das ContentLoaded DOM-Ereignis wird ausgelöst, wenn das ursprüngliche HTML-Dokument vollständig geladen und analysiert wurde. Zu diesem Zeitpunkt hat das Haupt-HTML-Dokument die Analyse abgeschlossen, der Browser beginnt mit der Erstellung der Rendering-Baumstruktur und untergeordnete Ressourcen müssen noch geladen werden. Es wird nicht drauf gewartet, bis Stylesheets, Bilder und Subframes vollständig geladen sind.	Stufe 1: `entryData.domContentLoadedEventStart > 0 ? entryData.domContentLoadedEventStart - entryData.navigationStart : 0` Ebene 2: EntryData.domContentLoadedEventStart

Name	Typ	Beschreibung	Hinweise
domContentLoaded	Zahl	Die Start- und Endzeit der Erstellung der Rendering-Baumstruktur ist gekennzeichnet durch `domContentLoadedEventStart` und `domContentLoadedEventEnd` . Damit kann CloudWatch RUM die Ausführung verfolgen. Diese Eigenschaft ist die Differenz zwischen `domContentLoadedStart` und `domContentLoadedEnd` . Während dieser Zeit sind DOM und CSSOM bereit. Diese Eigenschaft wartet auf die Ausführung des Skripts, mit Ausnahme von asynchronen und dynamisch erstellten Skripten. Wenn die Skripten von Stylesheets abhängen, wartet `domContentLoaded` auch auf die Stylesheets. Es wartet nicht auf Bilder. **Note** Die tatsächlichen Werte von `domContentLoadedStart` und `domContentLoadedEnd` nähern sich `domContentLoaded` im Netzwerkbereich von Google Chrome an. Es gibt die Erstellungszeit für die HTML-DOM+CSSOM-Rendering-Baumstruktur ab Beginn des Seitenladevorgangs an. Im Fall von Navigationsmetriken stellt der `domContentLoaded` -Wert die Differenz zwischen Start- und Endwerten dar. Dies ist ausschließlich die für das Herunterladen von untergeordneten Ressource	Ebene 2: EntryData.domContentLoadedEventEnd - Eingabedaten. domContentLoadedEventStart Stufe 2: EntryData.domContentLoadedEventEnd - Eingabedaten. domContentLoadedEventStart

Name	Typ	Beschreibung	Hinweise
		n und das Erstellen der Rendering-Baumstruktur benötigte Zeit.	
domComplete	Zahl	Die Zeit unmittelbar bevor der Browser die aktuelle Dokumentbereitschaft des aktuellen Dokuments als abgeschlossen festlegt. An diesem Punkt ist das Laden von untergeordneten Ressourcen wie Bildern abgeschlossen. Dies beinhaltet die Zeit, die für das Herunterladen blockierender Inhalte wie CSS und synchroner JavaScript Inhalte benötigt wird. Dies entspricht ungefähr `loadTime` im Netzwerkbereich von Google Chrome.	Stufe 1: `entryData.domComplete > 0 ? entryData.domComplete - entryData.navigationStart : 0` Stufe 2: entryData.domComplete

Name	Typ	Beschreibung	Hinweise
domProcessingTime	Zahl	Die Gesamtzeit zwischen der Antwort und dem Start des Load-Ereignisses.	Ebene 1: EntryData . loadEventStart - EntryData .ResponseEnd Ebene 2: EntryData . loadEventStart - EntryData .ResponseEnd
loadEventStart	Zahl	Die Zeit unmittelbar bevor das `load`-Ereignis des aktuellen Dokuments ausgelöst wird.	Stufe 1: `entryData .loadEventStart > 0 ? entryData .loadEventStart - entryData .navigationStart : 0` Ebene 2: EntryData . loadEventStart

Name	Typ	Beschreibung	Hinweise
loadEvent Time	Zahl	Differenz zwischen `loadEventStart` und `loadEventEnd` . Zusätzliche Funktionen oder Logik, die auf dieses Load-Ereignis warten, werden während dieser Zeit ausgelöst.	Ebene 1: EntryData . loadEvent End - Eingabeda ten. loadEvent Start Stufe 2: EntryData . loadEvent End - Eingabeda ten. loadEvent Start
duration	String	Die Dauer entspricht der gesamten Ladezeit der Seite. Es zeichnet den Zeitpunkt für das Herunterladen der Hauptseite und aller synchronen untergeordneten Ressource n sowie für das Rendern der Seite auf. Asynchrone Ressourcen wie Skripte werden später weiter heruntergeladen. Das ist die Differenz zwischen den Eigenschaften `loadEventEnd` und `startTime` .	Ebene 1: EntryData . loadEvent End - EntryData .Navigati onStart Stufe 2: entryData .duration

Name	Typ	Beschreibung	Hinweise
headerSize	Zahl	Gibt die Differenz zwischen `transferSize` und `encodedBodySize` zurück. Dieses Attribut ist nicht erforderlich.	Stufe 1: Nicht verfügbar Ebene 2: EntryData .TransferSize - EntryData . encodedBodySize Ebene 2: EntryData .TransferSize - EntryData . encodedBodySize
compressionRatio	Zahl	Das Verhältnis von `encodedBodySize` und `decodedBodySize` . Der Wert von `encodedBodySize` ist die komprimierte Größe der Ressource ohne die HTTP-Header. Der Wert von `decodedBodySize` ist die entkomprimierte Größe der Ressource ohne die HTTP-Header. Dieses Attribut ist nicht erforderlich.	Stufe 1: Nicht verfügbar. Stufe 2: `entryData.encodedBodySize > 0 ? entryData.decodedBodySize / entryData.encodedBodySize : 0`
navigationTimingLevel	Zahl	Die API-Version des Navigationszeitpunkts.	Wert: 1 oder 2

Ressourcen-Ereignisschema

Ressourcen-Ereignisse werden nur erfasst, wenn die App-Überwachung die Leistungstelemetrie aktiviert hat.

Timestamp-Metriken basieren auf The DOM HighResTimeStamp typedef. Bei APIs der Stufe 2 sind alle Timings standardmäßig relativ zur `startTime`. Aber für APIs der Stufe 1 wird die `navigationStart`-Metrik von Zeitstempelmetriken subtrahiert, um relative Werte zu erhalten. Alle Zeitstempelwerte sind in Millisekunden.

Vom Agent generierte Ressourcen-Ereignisse enthalten die folgenden Eigenschaften.

Name	Typ	Beschreibung	Hinweise
targetUrl	String	Gibt die URL der Ressource zurück.	Formel: entryData .name
initiatorType	String	Repräsentiert den Ressourcentyp, der das Leistungs-Ressourcenereignis ausgelöst hat.	Wert: „Ressource“ Formel: entryData .initiatorType
duration	String	Gibt die Differenz zwischen den Eigenschaften `responseEnd` und `startTime` zurück. Dieses Attribut ist nicht erforderlich.	Formel: entryData .duration
transferSize	Zahl	Gibt die Größe (in Oktetten) der abgerufenen Ressource zurück, einschließlich der Antworthe ader-Felder und des Antwort-Nutzlasttextes. Dieses Attribut ist nicht erforderlich.	Formel: entryData .transferSize
fileType	String	Erweiterungen, die vom Ziel-URL-Muster abgeleitet werden.	

Größtes inhaltliches Zeichnungsereignisschema

Die größten inhaltlichen Zeichnungsereignisse enthalten die folgenden Eigenschaften.

Diese Ereignisse werden nur erfasst, wenn die App-Überwachung die Leistungstelemetrie aktiviert hat.

Name	Beschreibung		
Wert	Weitere Informationen finden Sie unter Web Vitals (Webdaten).		

Erstes Eingabeverzögerungsereignis

Erste Eingabeverzögerungsereignisse enthalten die folgenden Eigenschaften.

Diese Ereignisse werden nur erfasst, wenn die App-Überwachung die Leistungstelemetrie aktiviert hat.

Name	Beschreibung		
Wert	Weitere Informationen finden Sie unter Web Vitals (Webdaten).		

Kumulatives Layoutverschiebungs-Ereignis

Kumulative Layoutverschiebungs-Ereignisse enthalten die folgenden Eigenschaften.

Diese Ereignisse werden nur erfasst, wenn die App-Überwachung die Leistungstelemetrie aktiviert hat.

Name	Beschreibung		
Wert	Weitere Informationen finden Sie unter Web Vitals (Webdaten).		

HTTP-Ereignis

HTTP-Ereignisse enthalten die folgenden Eigenschaften. Sie enthalten entweder ein `Response`-Feld oder ein `Error`-Feld, aber nicht beides.

Diese Ereignisse werden nur erfasst, wenn die App-Überwachung die HTTP-Telemetrie aktiviert hat.

Name	Beschreibung
Anforderung	Das Anfragefeld enthält Folgendes: • Das `Method`-Feld, das Werte wie GET, POST usw. haben kann. • Die URL
Antwort	Das Antwortfeld enthält die folgenden Elemente: • Status wie 2xx, 4xx oder 5xx • Statustext
Fehler	Das Fehlerfeld enthält die folgenden Elemente: • Typ • Fehlermeldung • Dateiname • Zeilennummer • Spaltennummer • Stack-Ablaufverfolgung

X-Ray-Ablaufverfolgungs-Ereignisschema

Diese Ereignisse werden nur erfasst, wenn die App-Überwachung die X-Ray-Ablaufverfolgungs-Telemetrie aktiviert hat.

Weitere Informationen zu Ereignisschemas für die X-Ray-Ablaufverfolgung finden Sie unter AWS X-Ray -Segmentdokumente.

Routenänderungs-Timing für Single-Page-Anwendungen

Wenn ein Benutzer in einer herkömmlichen mehrseitigen Anwendung das Laden neuer Inhalte anfordert, fordert der Benutzer eigentlich eine neue HTML-Seite vom Server an. Infolgedessen erfasst der CloudWatch RUM-Webclient die Ladezeiten mithilfe der regulären Performance-API-Metriken.

Einseitige Webanwendungen verwenden JavaScript jedoch Ajax, um die Benutzeroberfläche zu aktualisieren, ohne eine neue Seite vom Server zu laden. Single-Page-Aktualisierungen werden nicht von der Browser-Timing-API erfasst, sondern verwenden stattdessen das Routenänderungs-Timing.

CloudWatch RUM unterstützt die Überwachung sowohl ganzer Seitenladevorgänge vom Server als auch einzelner Seitenaktualisierungen mit den folgenden Unterschieden:

- Für das Routenänderungs-Timing gibt es keine browserseitig bereitgestellten Metriken wie `tlsTime` und `timeToFirstByte`.
- Beim Routenänderungs-Timing hat das Feld `initiatorType` den Wert `route_change`.

Der CloudWatch RUM-Webclient überwacht Benutzerinteraktionen, die zu einer Änderung der Route führen können, und wenn eine solche Benutzerinteraktion aufgezeichnet wird, zeichnet der Webclient einen Zeitstempel auf. Dann beginnt das Routenänderungs-Timing, wenn die beiden folgenden Punkte zutreffen:

- Für die Routenänderung wurde eine Browserverlaufs-API verwendet (mit Ausnahme der Browserschaltflächen für „Weiter“ und „Zurück“).
- Der zeitliche Abstand zwischen dem Zeitpunkt der Routenänderungserkennung und dem Zeitstempel der letzten Benutzerinteraktion beträgt weniger als 1 000 ms. Dadurch werden Datenverzerrungen vermieden.

Das gestartete Routenänderungs-Timing wird abgeschlossen, wenn keine laufenden AJAX-Anforderungen und DOM-Mutationen vorhanden sind. Anschließend wird der Zeitstempel der letzten abgeschlossenen Aktivität als Abschlusszeitstempel verwendet.

Für das Routenänderungs-Timing tritt ein Timeout auf, wenn AJAX-Anforderungen oder DOM-Mutationen länger als 10 Sekunden dauern (Standardeinstellung). In diesem Fall zeichnet der CloudWatch RUM-Webclient das Timing für diese Routenänderung nicht mehr auf.

Die Dauer eines Routenänderungsereignisses wird somit wie folgt berechnet:

```
(time of latest completed activity) - (latest user interaction timestamp)
```

Verwalten Sie Ihre Anwendungen, die CloudWatch RUM verwenden

Verwenden Sie die Schritte in diesen Abschnitten, um die Verwendung von CloudWatch RUM durch Ihre Anwendungen zu verwalten.

Wie finde ich einen Codeausschnitt, den ich bereits generiert habe?

Gehen Sie wie folgt vor, CloudWatch um einen RUM-Codeausschnitt zu finden, den Sie bereits für eine Anwendung generiert haben.

Bereits generierten Codeausschnitt finden

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, RUM aus.
3. Klicken Sie auf List view (Listenansicht).
4. Wählen Sie neben dem Namen des App-Monitors die Option Ansicht aus JavaScript.
5. Wählen Sie im Bereich JavaScript Snippet die Option In die Zwischenablage kopieren aus.

Bearbeiten Ihrer Anwendung

Gehen Sie folgendermaßen vor, um die Einstellungen einer App-Überwachung zu ändern. Sie können alle Einstellungen mit Ausnahme des Namens der App-Überwachung ändern.

Um zu bearbeiten, wie Ihre Anwendung RUM verwendet CloudWatch

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

2. Wählen Sie im Navigationsbereich Application Signals, RUM aus.
3. Klicken Sie auf List view (Listenansicht).
4. Wählen Sie die Schaltfläche neben dem Namen der Anwendung aus und wählen Sie dann Actions (Aktionen), Edit (Bearbeiten) aus.
5. Sie können alle Einstellungen mit Ausnahme des Namens der Anwendung ändern. Weitere Informationen zu diesen Einstellungen finden Sie unter Schritt 2: Erstellen Sie einer App-Überwachung.
6. Wenn Sie fertig sind, wählen Sie Speichern aus.

   Durch das Ändern der Einstellungen wird der Codeausschnitt geändert. Sie müssen jetzt den aktualisierten Codeausschnitt in Ihre Anwendung einfügen.
7. Nachdem der JavaScript Codeausschnitt erstellt wurde, wählen Sie In die Zwischenablage kopieren oder Herunterladen und dann Fertig aus.

   Fügen Sie den Codeausschnitt in Ihre Anwendung ein, um mit der Überwachung mit den neuen Einstellungen zu beginnen. Fügen Sie den Codeausschnitt in das `<head>`-Element Ihrer Anwendung vor dem `<body>`-Element oder anderen `<script>`-Tags ein.

Beenden Sie die Verwendung von CloudWatch RUM oder löschen Sie einen App-Monitor

Um die Verwendung von CloudWatch RUM mit einer Anwendung zu beenden, entfernen Sie den Codeausschnitt, den RUM aus dem Code Ihrer Anwendung generiert hat.

Gehen Sie folgendermaßen vor, um eine RUM-App-Überwachung zu löschen.

App-Überwachung löschen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Application Signals, RUM aus.
3. Klicken Sie auf List view (Listenansicht).
4. Wählen Sie die Schaltfläche neben dem Namen der Anwendung aus und wählen Sie dann Actions (Aktionen), Delete (Löschen) aus.
5. Geben Sie in das Bestätigungsfeld **`Delete`** ein und wählen Sie dann Delete (Löschen).
6. Wenn Sie dies noch nicht getan haben, löschen Sie den CloudWatch RUM-Codeausschnitt aus dem Code Ihrer Anwendung.

CloudWatch RUM-Kontingente

CloudWatch Für RUM gelten die folgenden Kontingente.

Ressource	Standardkontingent
App-Überwachungen	20 pro Konto Sie können eine Kontingenterhöhung beantragen.
RUM-Erfassungsrate	50 PutRumEventsAnfragen pro Sekunde (TPS). Sie können eine Kontingenterhöhung beantragen.

Problembehandlung bei RUM CloudWatch

Dieser Abschnitt enthält Tipps zur Behebung von Problemen mit CloudWatch RUM.

Es gibt keine Daten für meine Bewerbung

Stellen Sie zunächst sicher, dass der Codeausschnitt korrekt in Ihre Anwendung eingefügt wurde. Weitere Informationen finden Sie unter Schritt 4: Fügen Sie den Codeausschnitt in Ihre Anwendung ein.

Wenn dies nicht das Problem ist, gab es möglicherweise noch keinen Datenverkehr zu Ihrer Anwendung. Generieren Sie etwas Datenverkehr, indem Sie auf Ihre Anwendung genauso zugreifen wie ein Benutzer.

Für meine Anwendung wurden keine Daten mehr aufgezeichnet

Ihre Anwendung wurde möglicherweise aktualisiert und enthält jetzt keinen CloudWatch RUM-Codeausschnitt mehr. Überwachen Sie Ihren Anwendungscode.

Eine andere Möglichkeit besteht darin, dass jemand den Codeausschnitt aktualisiert, ihn aber nicht in die Anwendung eingefügt hat. Finden Sie den aktuell korrekten Codeausschnitt, indem Sie den Anweisungen unter Wie finde ich einen Codeausschnitt, den ich bereits generiert habe? folgen und ihn mit dem Codeausschnitt vergleichen, der in Ihre Anwendung eingefügt ist.

Netzwerk-Überwachung

Die Themen in diesem Abschnitt beschreiben die von Amazon Internet Monitor und Amazon CloudWatch Network Monitor bereitgestellten CloudWatch Netzwerk- und CloudWatch Internetüberwachungsfunktionen. Diese Services helfen Ihnen dabei, betriebliche Einblicke in die Netzwerk- und Internetleistung und Verfügbarkeit Ihrer Anwendungen zu erhalten, auf denen Sie gehostet AWS werden.

- Internet Monitor verwendet die Konnektivitätsdaten, die er aus seinem globalen Netzwerknetz AWS erfasst, um eine Ausgangsbasis für Leistung und Verfügbarkeit für den mit dem Internet verbundenen Datenverkehr zu berechnen. Sie können sich eine globale Ansicht der Verkehrsmuster und Gesundheitsereignisse anzeigen lassen und auf einfache Weise Informationen zu Ereignissen abrufen. Sie können sich auch Benachrichtigungen über Internet-Integritätsereignisse, die sich auf Ihre Anwendungsclients auswirken, erhalten. Darüber hinaus können Sie die Erkenntnisse, die Internet Monitor bietet, nutzen, um mögliche Verbesserungen Ihres Kundenerlebnisses zu untersuchen, indem Sie Amazon verwenden CloudFront oder über verschiedene Kanäle weiterleiten AWS-Regionen.
- Network Monitor verwendet einen vollständig verwalteten Agentenansatz, mit dem Sie Latenz und Paketverlust bei hybriden Netzwerkverbindungen verfolgen und visualisieren können. Um Messwerte zu sammeln und Network Monitor in die Lage zu versetzen, Warnmeldungen für Integritätsereignisse für Ihre Anwendung zu erstellen, erstellen Sie Sonden, die von Ihren gehosteten Ressourcen AWS an lokale Ziel-IP-Adressen gesendet werden. Sie müssen keine zusätzlichen Agenten installieren, um Ihre Netzwerkleistung zu überwachen. Wie bei Internet Monitor können Sie Warnmeldungen und Schwellenwerte festlegen, Informationen abrufen, die Ihnen helfen, Probleme schnell zu beheben, und dann Maßnahmen ergreifen, um Ihre Benutzererfahrung zu verbessern.

Themen

Amazon CloudWatch Internet Monitor verwenden

Amazon CloudWatch Internet Monitor gibt Aufschluss darüber, wie sich Internetprobleme auf die Leistung und Verfügbarkeit zwischen Ihren gehosteten Anwendungen AWS und Ihren Endbenutzern

auswirken. Dadurch kann die Zeit, die Sie für die Diagnose von Internetproblemen benötigen, von Tagen auf Minuten reduziert werden. Internet Monitor verwendet die Verbindungsdaten, die aus seiner globalen Netzwerkauslastung AWS erfasst werden, um eine Ausgangsbasis für Leistung und Verfügbarkeit für den mit dem Internet verbundenen Datenverkehr zu berechnen. Dabei handelt es sich um dieselben Daten, die zur Überwachung der Verfügbarkeit und Verfügbarkeit des Internets AWS verwendet werden. Auf der Grundlage dieser Messungen macht Internet Monitor Sie darauf aufmerksam, wenn bei Ihren Endbenutzern (Clients) an den verschiedenen geografischen Standorten, an denen Ihre Anwendung läuft, erhebliche Probleme auftreten.

In der CloudWatch Amazon-Konsole können Sie sich eine globale Ansicht der Verkehrsmuster und Gesundheitsereignisse anzeigen lassen und auf einfache Weise Informationen zu Ereignissen mit unterschiedlichen geografischen Granularitäten (Standorten) abrufen. Sie können die Auswirkungen deutlich visualisieren und die betroffenen Kundenstandorte und Netzwerke (ASNs, in der Regel Internetdienstanbieter oder ISPs) genau bestimmen. Wenn Internet Monitor feststellt, dass ein Problem mit der Internetverfügbarkeit oder der Leistung durch eine bestimmte ASN oder durch das AWS Netzwerk verursacht wird, werden diese Informationen bereitgestellt.

Hauptfeatures von Internet Monitor

- Internet Monitor schlägt Erkenntnisse und Empfehlungen vor, die Ihnen helfen können, die Erfahrung Ihrer Endnutzer zu verbessern. Sie können nahezu in Echtzeit herausfinden, wie Sie die erwartete Latenz Ihrer Anwendung verbessern können, indem Sie zu anderen Diensten wechseln oder den Datenverkehr über andere Dienste auf Ihren Workload umleiten. AWS-Regionen
- Mit Internet Monitor können Sie schnell erkennen, was sich auf die Leistung und Verfügbarkeit Ihrer Anwendung auswirkt, sodass Sie Probleme aufspüren und beheben können.
- Internet Monitor veröffentlicht Internet-Messwerte in CloudWatch Logs and CloudWatch Metrics, um die Verwendung von CloudWatch Tools mit Gesundheitsinformationen für Standorte und ASNs (Internetdienstanbieter) zu unterstützen, die für Ihre Anwendung spezifisch sind. Optional können Sie auch Internetmessungen in Amazon S3 veröffentlichen.
- Internet Monitor sendet Gesundheitsereignisse an Amazon, EventBridge sodass Sie Benachrichtigungen einrichten können. Wenn ein Problem durch das AWS Netzwerk verursacht wird, erhalten Sie außerdem automatisch eine AWS Health Dashboard Benachrichtigung mit den Maßnahmen, die AWS zur Behebung des Problems ergriffen wurden.

Verwendung von Internet Monitor

Um Internet Monitor zu verwenden, erstellen Sie einen Monitor und ordnen ihm die Ressourcen Ihrer Anwendung zu (VPCs, Network Load Balancer, CloudFront Distributionen oder WorkSpaces Verzeichnisse), damit Internet Monitor weiß, wo sich der mit dem Internet verbundene Datenverkehr Ihrer Anwendung befindet. Internet Monitor veröffentlicht dann Internet-Messwerte, die spezifisch für AWS die Stadtnetzwerke sind, d. h. die Client-Standorte und ASNs (in der Regel Internetdienstanbieter oder ISPs), über die Clients auf Ihre Anwendung zugreifen. Weitere Informationen finden Sie unter So funktioniert Amazon CloudWatch Internet Monitor. Um die Arbeit mit Internet Monitor zu beginnen, siehe Erste Schritte mit Amazon CloudWatch Internet Monitor mithilfe der Konsole.

Inhalt

Unterstützt AWS-Regionen für Amazon CloudWatch Internet Monitor

In diesem Abschnitt sind die Länder aufgeführt, in AWS-Regionen denen Amazon CloudWatch Internet Monitor unterstützt wird. Eine aktuelle Liste der Regionen, in denen Internet Monitor

unterstützt wird, einschließlich optionaler Regionen, finden Sie unter Amazon CloudWatch Internet Monitor-Endpunkte und Kontingente in der Amazon Web Services General Reference.

Beachten Sie, dass Internet Monitor Daten für einen Monitor nur in dem Land speichert, AWS-Region in dem Sie den Monitor erstellen, obwohl ein Monitor Ressourcen in mehreren Regionen enthalten kann.

Name der Region (Opt-In-Unterstützung)	Region
Afrika (Kapstadt)	`af-south-1`
Asien-Pazifik (Hongkong)	`ap-east-1`
Asien-Pazifik (Hyderabad)	`ap-south-2`
Asien-Pazifik (Jakarta)	`ap-southeast-3`
Asien-Pazifik (Melbourne)	`ap-southeast-4`
Europa (Milan)	`eu-south-1`
Europa (Spain)	`eu-south-2`
Europa (Zürich)	`eu-central-2`
Naher Osten (Bahrain)	`me-south-1`
Naher Osten (VAE)	`me-central-1`

Regionsname (Standardunterstützung)	Region
USA Ost (Ohio)	`us-east-2`
USA Ost (Nord-Virginia)	`us-east-1`
USA West (Nordkalifornien)	`us-west-1`
USA West (Oregon)	`us-west-2`
Asien-Pazifik (Mumbai)	`ap-south-1`

Regionsname (Standardunterstützung)	Region
Asia Pacific (Osaka)	ap-northeast-3
Asia Pacific (Seoul)	ap-northeast-2
Asien-Pazifik (Singapur)	ap-southeast-1
Asien-Pazifik (Sydney)	ap-southeast-2
Asien-Pazifik (Tokio)	ap-northeast-1
Canada (Central)	ca-central-1
Europe (Frankfurt)	eu-central-1
Europa (Irland)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Europa (Stockholm)	eu-north-1
Südamerika (São Paulo)	sa-east-1

Preise für Amazon CloudWatch Internet Monitor

Mit Amazon CloudWatch Internet Monitor fallen keine Vorabkosten oder langfristigen Verpflichtungen an. Die Preisgestaltung für Internet Monitor besteht aus zwei Komponenten: einer Gebühr pro überwachter Ressource und einer Stadtnetzgebühr. Ein Stadtnetz ist der Ort, von dem aus Kunden auf Ihre Anwendungsressourcen zugreifen, und das Netzwerk (ASN, z. B. ein Internetdienstanbieter oder ISP), über das die Kunden auf die Ressourcen zugreifen. Beachten Sie, dass Ihnen auch CloudWatch Standardpreise für Protokolle und alle zusätzlichen Metriken, Dashboards, Alarme oder Erkenntnisse, die Sie erstellen, berechnet werden.

Bei der Erstellung eines Monitors wählen Sie einen Prozentsatz des zu überwachenden Datenverkehrs aus. Um Ihre Kosten zu kontrollieren, können Sie auch ein Limit für die maximale Anzahl der zu überwachenden Stadtnetze festlegen. Sie können den Prozentsatz des zu überwachenden Datenverkehrs oder die maximale Anzahl der Stadtnetze jederzeit aktualisieren,

indem Sie Ihren Monitor bearbeiten. Die ersten 100 Stadtnetze (für alle Monitore pro Konto) sind enthalten. Danach zahlen Sie nur für die tatsächliche zusätzliche Anzahl von Stadtnetzen, die Sie überwachen, bis zur maximalen Anzahl.

Sie zahlen nur die tatsächliche zusätzliche Anzahl der von Ihnen überwachten Stadtnetze, bis zur maximalen Anzahl, wobei die ersten 100 Stadtnetze (über alle Monitore pro Konto) kostenlos sind. Ein Pauschalbetrag, der den Kosten für 100 Stadtnetze entspricht, wird von Ihrer monatlichen Rechnung abgezogen.

Ein großes globales Unternehmen könnte sich beispielsweise dafür entscheiden, 100 % seines Internetverkehrs zu überwachen und ein Maximum von 50 000 Stadtnetzen für einen Monitor mit einer Ressource festzulegen. Angenommen, der Datenverkehr erreicht 50 000 Stadtnetze, dann würde dieser Teil der Rechnung etwa 2 700 USD/Monat betragen. Für ein anderes Unternehmen in weniger geografischen Gebieten, mit einem Monitor mit einer Ressource und 200 Stadtnetzwerken würde sich dieser Teil der Rechnung auf etwa 13 USD pro Monat belaufen. Weitere Informationen finden Sie unter Auswahl einer Höchstgrenze für Städtenetze.

Mit dem Preisrechner können Sie verschiedene Optionen ausprobieren. Scrollen Sie auf der CloudWatch Seite Preisrechner für nach unten zu Internet Monitor, um mehr über die Preisoptionen zu erfahren.

Weitere Informationen zu Internet Monitor und CloudWatch Preisen finden Sie auf der CloudWatch Amazon-Preisseite.

Komponenten und Bedingungen für Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor verwendet oder verweist auf Folgendes.

Überwachen

Ein Monitor enthält die Ressourcen für eine einzelne Anwendung, für die Sie Internet-Leistungs- und Verfügbarkeitsmessungen anzeigen und Zustandsereignis-Warnungen erhalten möchten. Wenn Sie einen Monitor für eine Anwendung erstellen, fügen Sie Ressourcen für die Anwendung hinzu, um die Städte (Standorte) zu definieren, die Internet Monitor überwachen soll. Internet Monitor verwendet die Datenverkehrsmuster der von Ihnen hinzugefügten Anwendungsressourcen, um Internet-Leistungs- und Verfügbarkeitsmessungen zu veröffentlichen, die nur für die Standorte und ASNs (in der Regel Internetdienstanbieter oder ISPs) gelten, die mit Ihrer Anwendung kommunizieren. Mit anderen Worten: Die von Ihnen hinzugefügten Ressourcen bilden den Umfang der Stadtnetze, die Internet Monitor überwachen soll und für die Messungen veröffentlicht werden sollen.

Dem Monitor hinzugefügte Ressource („überwachte Ressource“)

Eine Ressource, die Sie einem Monitor hinzufügen, ist in Internet Monitor eine „überwachte Ressource“. Das heißt:

- Jede VPC, die Sie einer Region hinzufügen, ist eine überwachte Ressource. Wenn Sie eine VPC hinzufügen, überwacht Internet Monitor den Datenverkehr für jede mit dem Internet verbundene Anwendung in der VPC, z. B. eine Anwendung, die auf einer Amazon EC2 EC2-Instance, hinter einem Network Load Balancer oder einem Container gehostet wird. AWS Fargate
- Jeder Network Load Balancer, den Sie einer Region hinzufügen, ist eine überwachte Ressource.
- Jedes WorkSpaces Verzeichnis, das Sie einer Region hinzufügen, ist eine überwachte Ressource.
- Jede CloudFront Distribution, die Sie hinzufügen, ist eine überwachte Ressource.

Autonome Systemnnummer (ASN)

In Internet Monitor bezieht sich eine ASN in der Regel auf einen Internetdienstanbieter (ISP) wie zum Beispiel Verizon oder Comcast. Eine ASN ist ein Netzwerkanbieter, über den ein Client auf Ihre Internetanwendung zugreift. Ein autonomes System (AS) ist eine Gruppe von Internet-Routing-Präfixen (IP-Präfixe), die zu einem Netzwerk oder einer Sammlung von Netzwerken gehören, die alle von einer Organisation verwaltet, kontrolliert und überwacht werden.

Stadtnetz (Standort und ASN)

Ein Stadtnetzwerk ist der Standort (z. B. eine Stadt), von dem aus Clients auf Ihre Anwendungsressourcen zugreifen, und die ASN, in der Regel ein Internetdienstanbieter (ISP), über die die Clients auf die Ressourcen zugreifen. Um Ihre Rechnung unter Kontrolle zu halten, können Sie ein Limit für die maximale Anzahl von Stadtnetzwerken festlegen, die Internet Monitor für jeden Monitor überwachen soll. Sie zahlen nur für die tatsächliche Anzahl der Stadtnetze, die Sie überwachen, und zwar bis zur maximalen Anzahl. Weitere Informationen finden Sie unter Auswählen einer Höchstgrenze für Stadtnetze.

Internet-Messungen

Internet Monitor veröffentlicht alle fünf Minuten Internet-Messwerte in Protokolldateien in CloudWatch Logdateien für die 500 wichtigsten Stadtnetzwerke (Kundenstandorte und ASNs, in der Regel Internetdienstanbieter oder ISPs) in Ihrem Konto. Diese Messungen quantifizieren den Leistungswert, den Verfügbarkeitswert, die übertragenen Bytes (eingehende und ausgehende Bytes) und die Round-Trip-Zeit für die Stadtnetze Ihrer Anwendung. Dies sind Messungen für die

Stadtnetzwerke, die für Ihre VPCs, Network Load Balancer, Distributionen oder Verzeichnisse spezifisch sind. CloudFront WorkSpaces Optional können Sie Internetmessungen und Ereignisse für alle überwachten Stadtnetze (bis zum Limit von 500 000 Stadtnetzen) in einem Amazon-S3-Bucket veröffentlichen.

Metriken

Internet Monitor generiert aggregierte Metriken für CloudWatch Metriken, für den globalen Datenverkehr zu Ihrer Anwendung und für den globalen Datenverkehr zu jeder Anwendung. AWS-Region Weitere Informationen finden Sie unter Verwenden von CloudWatch Metriken mit Amazon CloudWatch Internet Monitor.

Zustandsereignis

Internet Monitor erstellt ein Zustandsereignis, um Sie auf ein bestimmtes Problem aufmerksam zu machen, das Ihre Anwendung betrifft. Internet Monitor erkennt Internetprobleme, wie z. B. erhöhte Netzwerklatenz, weltweit. Anschließend berechnet er anhand seiner historischen Internet-Messwerte aus der gesamten AWS globalen Infrastruktur die Auswirkungen aktueller Probleme auf Ihre Anwendung und generiert Gesundheitsereignisse. Internet Monitor erstellt standardmäßig Zustandsereignisse auf der Grundlage von Schwellenwerten für die Gesamtauswirkungen und die lokalen Auswirkungen. Weitere Informationen zur Konfiguration von Schwellenwerten finden Sie unter Schwellenwerte für Zustandsereignisse ändern.

Jedes Zustandsereignis enthält Informationen über die betroffenen Städtenetze. Sie können Integritätsereignisse in der CloudWatch Konsole, mithilfe des AWS SDK oder AWS CLI mithilfe von Internet Monitor-API-Aktionen anzeigen. Internet Monitor sendet auch EventBridge Amazon-Benachrichtigungen über Gesundheitsereignisse. Weitere Informationen finden Sie unter Wann erstellt Internet Monitor Zustandsereignisse und löst sie auf.

Internet-Ereignis

Internet Monitor zeigt Informationen über aktuelle globale Gesundheitsereignisse, sogenannte Internetereignisse, auf einer Internet-Wetterkarte an, die allen AWS Kunden zur Verfügung steht. Sie müssen in Internet Monitor keinen Monitor erstellen, um die Internet-Wetterkarte anzuzeigen. Im Gegensatz zu Gesundheitsereignissen beziehen sich Internetereignisse nicht auf einzelne Kunden oder deren Anwendungsdatenverkehr. Weitere Informationen finden Sie unter Weltweite Internet-Wetterkarte in Amazon CloudWatch Internet Monitor.

Schwellenwerte

Internet Monitor erstellt Zustandsereignisse auf der Grundlage von allgemeinen und lokalen Schwellenwerten. Sie können die standardmäßigen Schwellenwerte ändern und andere Optionen

konfigurieren, wie z. B. die Deaktivierung lokaler Schwellenwerte. Weitere Informationen zur Konfiguration von Schwellenwerten finden Sie unter Schwellenwerte für Zustandsereignisse ändern.

Leistungs- und Verfügbarkeitswertungen

Durch die Analyse der AWS gesammelten Daten kann Internet Monitor im Vergleich zu den von Internet Monitor berechneten geschätzten Ausgangswerten erkennen, wann die Leistung und Verfügbarkeit Ihrer Anwendung nachgelassen hat. Damit Sie diese Rückgänge leichter erkennen können, meldet Internet Monitor die Informationen in Form von Werten an Sie. Eine Leistungsbewertung gibt den geschätzten Prozentsatz des Datenverkehrs an, bei dem kein Leistungsabfall zu verzeichnen ist. In ähnlicher Weise stellt eine Verfügbarkeitsbewertung den geschätzten Prozentsatz des Datenverkehrs dar, für den kein Verfügbarkeitsrückgang zu verzeichnen ist. Weitere Informationen finden Sie unter So werden Leistungs- und AWS Verfügbarkeitswerte berechnet.

Übertragene Bytes und überwachte übertragene Bytes

Die übertragenen Byte sind die Gesamtzahl der ein- und ausgehenden Datenverkehrs in Byte zwischen einer Anwendung und dem Stadtnetzwerk (d. h. dem Standort AWS und der ASN, in der Regel dem Internetdienstanbieter), über das Clients auf eine Anwendung zugreifen. Überwachte übertragene Bytes ist eine ähnliche Metrik, enthält aber nur Bytes für überwachten Verkehr.

Round-Trip-Time

Die Round-Trip-Zeit (RTT) gibt an, wie lange es dauert, bis eine Anfrage von einem Client-Benutzer eine Antwort an den Benutzer zurückgibt. Wenn die RTT nach Kundenstandorten (Städten oder anderen geografischen Gebieten) aggregiert wird, wird der Wert danach gewichtet, wie viel von Ihrem Anwendungsverkehr von jedem Kundenstandort stammt.

Weltweite Internet-Wetterkarte in Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor zeigt eine globale Internet-Wetterkarte an, die allen AWS Kunden zur Verfügung steht. Um die Karte anzuzeigen, navigieren Sie in der CloudWatch Amazon-Konsole zu Internet Monitor.

Auf der Karte werden Internetereignisse („Ausfälle“) auf der ganzen Welt hervorgehoben, die sich auf AWS Kunden auswirken, und zwar in den jeweiligen Städten und Netzwerken (ASNs, in der Regel Internetdienstanbieter), in denen Leistungs- oder Verfügbarkeitsprobleme auftreten. Die Internet-Wetterkarte enthält Internetereignisse der letzten 24 Stunden.

Sie müssen in Internet Monitor keinen Monitor erstellen, um die Internet-Wetterkarte anzuzeigen. Im Gegensatz zu Gesundheitsereignissen in Internet Monitor beziehen sich Internetereignisse nicht auf einzelne Kunden oder deren Anwendungsdatenverkehr.

Auf der Internet-Wetterkarte können Sie ein Internetereignis auswählen, um Einzelheiten darüber zu erfahren. Bei einer Internetveranstaltung können Sie die Startzeit, die Endzeit (falls die Veranstaltung vorbei ist), den aktuellen Status (Aktiv oder Behoben) und die Art des Ausfalls (Verfügbarkeit oder Leistung) sehen. Weitere Informationen darüber, wie die Internet-Wetterkarte erstellt wird und was sie beinhaltet, finden Sie in den häufig gestellten Fragen zu globalen Internet-Wetterkarten.

Um detaillierte Informationen zu Ihrem Anwendungsdatenverkehr und Ihren Client-Standorten anzuzeigen und mit ihnen zu arbeiten, können Sie in Internet Monitor ganz einfach einen Monitor für Ihre Anwendung einrichten. Anschließend sehen Sie aktuelle und historische Leistungs- und Verfügbarkeitsmuster und Ereignisse sowie Benachrichtigungen zu Integritätsereignissen, die genau auf Ihren Anwendungsbestand und Ihre Kunden zugeschnitten sind. Die Internet-Wetterkarte bietet Ihnen einen Gesamtüberblick, während ein spezieller Monitor die Informationen nur nach den Messungen und Details filtert, die für Ihre Anwendung relevant sind. Mit einem Monitor können Sie auch historische Kennzahlen untersuchen und Empfehlungen zur Verbesserung des Kundenerlebnisses für Ihre Anwendung erhalten. Weitere Informationen hierzu finden Sie unter Erste Schritte mit Amazon CloudWatch Internet Monitor mithilfe der Konsole.

So funktioniert Amazon CloudWatch Internet Monitor

Dieser Abschnitt enthält Informationen zur Funktionsweise von Amazon CloudWatch Internet Monitor. Darin wird beschrieben, wie die Daten AWS erfasst werden, anhand derer Verbindungsprobleme im Internet erkannt werden, und wie die Leistungs- und Verfügbarkeitswerte berechnet werden.

Inhalt

Wie sich Internet Monitor nur auf den Fußabdruck Ihres Anwendungsverkehrs konzentriert

Internet Monitor konzentriert sich bei der Überwachung nur auf den Teil des Internets, auf den die Benutzer Ihrer AWS Ressourcen zugreifen, anstatt Ihre Website von allen Regionen der Welt aus umfassend zu überwachen, wie dies bei anderen Tools der Fall ist. Es ist auch eine kostengünstige Lösung, die für große und kleine Unternehmen erschwinglich ist.

Internet Monitor verwendet dieselben leistungsstarken Tests und Algorithmen zur Problemerkennung, die auch intern genutzt werden, AWS und warnt Sie vor Verbindungsproblemen, die sich auf Ihre Anwendung auswirken, indem es Integritätsereignisse in Internet Monitor erstellt. Internet Monitor ermöglicht Ihnen dann den Zugriff auf die resultierende Leistungs- und Verfügbarkeitskarte, indem es das Datenverkehrsprofil überlagert, das es auf der Grundlage Ihrer Anwendungsressourcen von Ihren aktiven Betrachtern erstellt.

Anhand dieser Informationen zeigt Ihnen Internet Monitor nur relevante Ereignisse (d. h. die Ereignisse von Orten, an denen Sie aktive Betrachter haben) und nur die Auswirkungen dieser Ereignisse auf Ihr Gesamtbetrachtervolumen. Die prozentuale Auswirkung eines Ereignisses hängt also von Ihrem gesamten weltweiten Traffic ab.

Internet Monitor veröffentlicht in CloudWatch Logs alle fünf Minuten Internet-Messungen für die 500 wichtigsten Stadtnetzwerke (Kundenstandorte und ASNs, in der Regel Internetdienstanbieter oder ISPs), die Datenverkehr an die einzelnen Monitore senden. Optional können Sie Internetmessungen für alle überwachten Stadtnetze (bis zum Limit von 500 000 Stadtnetzen) in einem Amazon-S3-Bucket veröffentlichen. Weitere Informationen finden Sie unter Internet-Messungen in Amazon S3 in Amazon CloudWatch Internet Monitor veröffentlichen.

Internet Monitor bietet folgende Vorteile:

- Die Verwendung von Internet Monitor verursacht keine zusätzliche Belastung oder Kosten für Ihre Anwendung, die auf AWS gehostet wird.
- Sie müssen keinen Code zur Leistungsmessung in Ihre clientseitigen Ressourcen oder in Ihre Anwendung aufnehmen.
- Sie können sich einen Überblick über die Leistung und Verfügbarkeit im Internet verschaffen, mit dem Ihre Anwendung verbunden ist, inklusive Informationen zur „letzten Meile“.

Beachten Sie, dass Internet Monitor Messungen auf der Grundlage Ihrer AWS Ressourcen erstellt, sodass Internet Monitor nur Ereignisse generiert, die für Ihren Anwendungsdatenverkehr spezifisch sind. Globale Internetprobleme werden im Allgemeinen nicht gemeldet. Wenn es sich bei dem Servicestandort um einen handelt AWS-Region, sind die ausgesendeten Messungen und

Ereignisse außerdem so konzipiert, dass sie die Konnektivität auf regionaler Ebene darstellen und nicht genau die Konnektivität zwischen einem Endbenutzerstandort und einer Availability Zone darstellen.

Wie AWS misst Verbindungsprobleme und berechnet Messwerte

Amazon CloudWatch Internet Monitor verwendet Internetverbindungsdaten zwischen verschiedenen AWS-Regionen und CloudFront Amazon-Points of Presence (POPs) zu verschiedenen Kundenstandorten über autonome Systemnummern (ASNs), in der Regel Internetdienstanbieter (ISPs). Dies sind die Verbindungsdaten, die täglich intern von AWS den Betreibern verwendet werden, um Verbindungsprobleme im globalen Internet proaktiv zu erkennen.

Für jeden dieser Bereiche wissen wir AWS-Region, welche Teile des Internets mit der Region kommunizieren, und gehen wie folgt vor:

- Wir überwachen diese Teile des Internets aktiv mit einem fortlaufenden Zeitfenster von 30 Tagen.
- Wir verwenden sowohl Netzwerk- als auch Protokollprüfungen auf höherer Ebene, einschließlich eingehender und ausgehender Abfragen.

AWS verfügt über aktive und passive Tests, die die Latenz (Leistung) im 90. Perzentil und die Erreichbarkeit (Verfügbarkeit) von jedem AWS-Region Dienst bis zum gesamten Internet messen. CloudFront Abnormale Verbindungsmuster zwischen einem Service und einem Kundenstandort werden überwacht und anschließend als Warnmeldung an den Kunden gemeldet.

Berechnung von Verfügbarkeit und RTT

Die Roundtrip-Zeit (RTT) gibt an, wie lange es dauert, bis eine Anfrage des Benutzers eine Antwort an den Benutzer zurückgibt. Wenn die Roundtrip-Zeit für alle Endbenutzerstandorte aggregiert wird, wird der Wert anhand der Menge Ihres Datenverkehrs gewichtet, der von den einzelnen Endbenutzerstandorten generiert wird.

Beispiel: Bei zwei Endbenutzerstandorten, von denen einer 90 % des Datenverkehrs mit einer 5-ms-RTT und der andere 10 % des Datenverkehrs mit einer 10-ms-RTT abwickelt, ergibt sich eine aggregierte RTT von 5,5 ms (die sich aus 5 ms x 0,9 + 10 ms x 0,1 ergibt).

Beachten Sie, dass es bei den Ressourcen bei der Messung der Latenz auf der letzten Meile Unterschiede gibt. Bei Latenzmessungen von Internet Monitor berücksichtigen VPCs, Network Load Balancer und WorkSpaces Verzeichnisse keine Latenz auf der letzten Meile.

Berechnung der Leistungs- und Verfügbarkeitswerte

AWS verfügt über umfangreiche historische Daten zur Internetleistung und -verfügbarkeit zwischen AWS Diensten und verschiedenen Stadtnetzen (Standorte und ASNs). Indem Internet Monitor die Daten einer statistischen Analyse unterzieht, kann er feststellen, wann die Leistung und Verfügbarkeit Ihrer Anwendung im Vergleich zu einer geschätzten Baseline, die er berechnet hat, gesunken ist. Damit Sie diese Rückgänge leichter erkennen können, werden Ihnen diese Informationen in Form von Zustandsbewertungen mitgeteilt: eine Leistungsbewertung und eine Verfügbarkeitsbewertung.

Die Zustandsbewertungen werden mit unterschiedlichen Granularitäten berechnet. Mit feinster Granularität berechnen wir die Zustandsbewertung für eine geografische Region, z. B. eine Stadt oder eine Metropolregion, und einen ASN (ein Stadtnetz). Wir rechnen die einzelnen Zustandsbewertungen auch zu Gesamt-Zustandswerten für eine Anwendung in einem Monitor zusammen. Wenn Sie Leistungs- oder Verfügbarkeitsbewertungen anzeigen, ohne nach einer bestimmten Region oder einem bestimmten Dienstanbieter zu filtern, bietet Internet Monitor umfassende Zustandsbewertungen.

Die Gesamtbewertung des Zustands umfasst Ihre komplette Anwendung für die angegebene Zeitspanne. Wenn der Leistungs- oder Verfügbarkeitswert für die Stadtnetzpaare Ihrer Anwendung den entsprechenden Zustandsereignis-Schwellenwert für Leistung oder Verfügbarkeit erreicht oder unterschreitet, löst Internet Monitor ein Zustandsereignis aus. Standardmäßig liegt der Schwellenwert sowohl für die Gesamtleistung als auch für die Verfügbarkeit bei 95 %. Internet Monitor erstellt auch Zustandsereignisse auf der Grundlage lokaler Schwellenwerte – wenn diese Option aktiviert ist, was standardmäßig der Fall ist – und auf der Grundlage von Werten, die Sie konfigurieren. Weitere Informationen zur Konfiguration von Schwellenwerten für Zustandsereignisse finden Sie unter Schwellenwerte für Zustandsereignisse ändern.

Wenn Sie Informationen im Monitor und in den Protokolldateien untersuchen, um Probleme zu ermitteln und mehr zu erfahren, können Sie nach bestimmten Städten (Standorten), Netzwerken (ASNs oder Internetdienstanbietern) oder beidem filtern. Sie können also Filter verwenden, um Zustandswerte für verschiedene Städte, ASNs oder Stadtnetzpaare anzuzeigen, je nachdem, welche Filter Sie wählen.

- Eine Verfügbarkeitsbewertung stellt den geschätzten Prozentsatz des Datenverkehrs dar, für den kein Verfügbarkeitsrückgang zu verzeichnen ist. Internet Monitor schätzt den Prozentsatz des Datenverkehrs, bei dem ein Rückgang zu verzeichnen ist, ausgehend vom Gesamtdatenverkehr und den Messungen der Verfügbarkeitsmetriken. Beispielsweise

entspricht eine Verfügbarkeitsbewertung von 99 % für ein Paar aus Endbenutzer- und Servicestandort einem Verfügbarkeitsverlust von 1 % des Datenverkehrs für dieses Paar.

- Ein Leistungswert gibt den geschätzten Prozentsatz des Datenverkehrs an, bei dem kein Leistungsabfall zu verzeichnen ist. Beispielsweise entspricht ein Leistungswert von 99 % für ein Paar aus Endbenutzer- und Servicestandort einem Leistungsverlust von 1 % des Datenverkehrs für dieses Paar.

Berechnung von TTFB und RTT (Latenz)

Die Zeit bis zum ersten Byte (TTFB) bezieht sich auf die Zeit zwischen dem Zeitpunkt, an dem ein Client eine Anfrage stellt, und dem Empfang des ersten Informationsbytes vom Server. AWS Berechnungen für TTFB messen die Zeit, die von Amazon EC2 oder Amazon CloudFront bis zum Internet Monitor-Messknoten vergangen ist (einschließlich der letzten Meile des Knotens). Das heißt, Internet Monitor misst die Zeit vom Benutzer zur Amazon EC2 EC2-Region für TTFB für EC2 und vom Benutzer zur Amazon EC2-Region CloudFront für. CloudFront

Für die Round-Trip-Zeit (RTT) berücksichtigt Internet Monitor die Zeit vom Stadtnetz (also dem Standort des Kunden und dem ASN, in der Regel ein Internetdienstanbieter), wie sie durch die öffentliche IP-Adresse abgebildet wird, bis zur AWS-Region. Das bedeutet, dass Internet Monitor für Benutzer, die von hinter einem Gateway oder VPN auf das Internet zugreifen, keine Sichtbarkeit auf der letzten Meile bietet.

Beachten Sie, dass es bei den Ressourcen bei der Messung der Latenz auf der letzten Meile Unterschiede gibt. Bei Latenzmessungen von Internet Monitor berücksichtigen VPCs, Network Load Balancer und WorkSpaces Verzeichnisse keine Latenz auf der letzten Meile.

Internet Monitor enthält durchschnittliche TTFB-Informationen im Bereich Vorschläge zur Verkehrsoptimierung auf der Registerkarte Traffic Insights auf dem CloudWatch Dashboard, damit Sie Optionen für verschiedene Setups für Ihre Anwendung bewerten können, mit denen die Leistung verbessert werden kann.

Messungen und Aggregation nach Regionen und Verfügbarkeitszonen

Internet Monitor aggregiert zwar Messungen und teilt die Auswirkungen auf regionaler Ebene, berechnet die Auswirkungen jedoch auf der Ebene der Availability Zone (AZ). Das heißt, wenn bei einem Ereignis nur eine AZ betroffen ist und der Großteil Ihres Datenverkehrs durch diese AZ fließt, sehen Sie Auswirkungen auf Ihren Traffic. Wenn Ihr Anwendungsdatenverkehr bei demselben Ereignis nicht über eine betroffene AZ fließt, sehen Sie jedoch keine Auswirkungen.

Beachten Sie, dass dies nur für Ressourcen gilt, bei denen es sich nicht um WorkSpaces Verzeichnisse handelt. WorkSpaces Verzeichnisse werden nur auf regionaler Ebene gemessen.

Genauigkeit der Geolokalisierung in Internet Monitor

Für Standortinformationen verwendet Internet Monitor IP-Geolocation-Daten, die von bereitgestellt werden. MaxMind Die Genauigkeit der Standortinformationen in Internet Monitor-Messungen hängt von der Genauigkeit der MaxMind Daten ab.

Beachten Sie, dass die `Metro` Füllstandsmessungen für Standorte außerhalb der USA möglicherweise nicht genau sind.

Wann Internet Monitor Zustandsereignisse erstellt und auflöst

Internet Monitor erstellt und schließt Zustandsereignisse für den von Ihnen überwachten Anwendungsverkehr auf der Grundlage der aktuell eingestellten Schwellenwerte. Internet Monitor verfügt über eine Standardkonfiguration für Schwellenwerte, Sie können auch Ihre eigene Konfiguration für Schwellenwerte festlegen. Internet Monitor ermittelt die Gesamtauswirkungen von Konnektivitätsproblemen auf Ihre Anwendung sowie die Auswirkungen auf lokale Bereiche, in denen Ihre Anwendung Kunden hat, und erzeugt Zustandsereignisse, wenn die Schwellenwerte überschritten werden.

Internet Monitor berechnet die Auswirkungen von Verbindungsproblemen auf einen Client-Standort auf der Grundlage der historischen Daten zur Internetleistung und -verfügbarkeit für den Netzwerkverkehr, über die der Dienst verfügbar ist. AWS Er wendet die für Ihre Anwendung relevanten Informationen an, basierend auf den geografischen Standorten für ASNs und Services, an denen Kunden Ihre Anwendung nutzen: die betroffenen Stadtnetzpaare. Die Standorte werden anhand der Ressourcen bestimmt, die Sie Ihrem Monitor hinzufügen. Internet Monitor verwendet dann statistische Analysen, um festzustellen, wann die Leistung und die Verfügbarkeit gesunken sind, was sich auf das Kundenerlebnis für Ihre Anwendung auswirkt.

Die von Internet Monitor berechneten Leistungs- und Verfügbarkeitswerte werden als prozentualer Anteil des Datenverkehrs dargestellt, bei dem es zu keinem Datenverlust kommt. Auswirkungen sind das Gegenteil davon: sie repräsentieren, wie problematisch ein Problem für die Endbenutzer eines Kunden ist. Wenn es also beispielsweise zu einem globalen Rückgang der Verfügbarkeit um 93 % kommt, wären die entsprechenden Auswirkungen 7 %.

Wenn der Leistungs- oder Verfügbarkeitswert für die Stadtnetzpaare Ihrer Anwendung global den entsprechenden Schwellenwert für Leistung oder Verfügbarkeit erreicht oder unterschreitet,

löst dies bei Internet Monitor ein Zustandsereignis aus. Standardmäßig liegt der Schwellenwert sowohl für Leistung als auch für Verfügbarkeit bei 95 %. Die Werte für das Erreichen oder Unterschreiten des Schwellenwerts sind kumulativ, das heißt, es kann bedeuten, dass mehrere kleinere Ereignisse zusammen den Schwellenwert erreichen, oder dass ein einzelnes Ereignis den Schwellenwert erreicht oder unterschreitet.

Solange die Leistungs- oder Verfügbarkeitswerte, die das Ereignis ausgelöst haben, auf oder unter dem entsprechenden prozentualen Schwellenwert für Zustandsereignisse in Bezug auf die Gesamtauswirkungen liegen, bleibt das Zustandsereignis aktiv. Wenn die Werte oder die kombinierten Werte, die das Ereignis ausgelöst haben, über den Schwellenwert steigen, löst Internet Monitor das Zustandsereignis auf.

Internet Monitor erstellt auch Zustandsereignisse auf der Grundlage lokaler Schwellenwerte und des prozentualen Anteils am Gesamtverkehr, auf den sich ein Problem auswirkt. Sie können Optionen für lokale Schwellenwerte konfigurieren oder lokale Schwellenwerte ganz abschalten.

Weitere Informationen zur Konfiguration von Schwellenwerten für Zustandsereignisse finden Sie unter Schwellenwerte für Zustandsereignisse ändern.

Zeitpunkt der Meldung von Zustandsereignissen

Internet Monitor verwendet einen Aggregator, um alle Signale zu Internetproblemen zu sammeln und innerhalb weniger Minuten Zustandsereignisse in Überwachungen zu erstellen.

Wenn möglich, analysiert Internet Monitor den Ursprung eines Integritätsereignisses, um festzustellen, ob es durch AWS oder durch eine ASN verursacht wurde. Die Analyse von Zustandsereignissen wird fortgesetzt, nachdem ein Ereignis behoben wurde. Internet Monitor kann Ereignisse mit neuen Informationen bis zu eine Stunde lang aktualisieren.

So funktioniert Internet Monitor mit IPv4- und IPv6-Verkehr

Internet Monitor misst das Zustandsereignis für ein Netzwerk, das nur über IPv4 läuft, und zeigt Ihnen Zustandsereignisse sowie Verfügbarkeits- und Leistungsmetriken an, wenn Sie den Datenverkehr für dieses Netzwerk über eine beliebige IP-Familie (IPv4 oder IPv6) bereitstellen. Wenn Sie Datenverkehr von einer Dual-Stack-Ressource, z. B. einer CloudFront Dual-Stack-Verteilung, bereitstellen, löst Internet Monitor nur dann ein Integritätsereignis aus und zeigt einen Rückgang der Leistungs- oder Verfügbarkeitsbewertung an, wenn der IPv4-Verkehr dieselben Probleme für die Ressource hat wie der IPv6-Verkehr.

Beachten Sie, dass die Metriken von Internet Monitor für die Gesamtheit der eingehenden und ausgehenden Bytes den gesamten Internetverkehr (IPv4 und IPv6) genau wiedergeben.

Wie Internet Monitor die Teilmenge der Stadtnetzwerke auswählt, die einbezogen werden sollen

Wenn Sie eine Obergrenze für die Anzahl der von Ihrem Monitor überwachten Stadtnetzwerke festlegen oder einen Prozentsatz des zu überwachenden Datenverkehrs auswählen, wählt Internet Monitor die zu berücksichtigenden Stadtnetzwerke auf der Grundlage des höchsten aktuellen Verkehrsaufkommens aus (überwacht).

Wenn Sie beispielsweise eine Obergrenze für Stadtnetzwerke auf 100 festlegen, überwacht Internet Monitor (bis zu) 100 Stadtnetzwerke auf der Grundlage Ihres Anwendungsverkehrs während der letzten Stunde. Insbesondere überwacht Internet Monitor die 100 Stadtnetzwerke, die im letzten einstündigen Fenster vor dem letzten einstündigen Fenster den meisten Verkehr hatten.

Nehmen wir zur Veranschaulichung an, die aktuelle Uhrzeit ist 14:30 Uhr. In diesem Szenario wurde der Verkehr, den Sie auf Ihrem Monitor sehen, zwischen 13:00 Uhr und 14:00 Uhr erfasst, und die Messung des Verkehrsvolumens, anhand derer Internet Monitor die 100 wichtigsten Stadtnetzwerke ermittelt, wurde zwischen 12:00 Uhr und 13:00 Uhr erfasst.

Wie die globale Internet-Wetterkarte erstellt wird (Häufig gestellte Fragen)

Die CloudWatch Internet-Wetterkarte von Amazon Internet Monitor ist auf der Internet Monitor-Konsole für alle authentifizierten AWS Kunden verfügbar. Dieser Abschnitt enthält Einzelheiten darüber, wie die Internet-Wetterkarte erstellt wird und wie sie verwendet wird.

Was ist die Internet-Wetterkarte von Internet Monitor?

Die Internet-Wetterkarte bietet eine visuelle Darstellung von Internetproblemen auf der ganzen Welt. Sie hebt die Standorte der betroffenen Kunden hervor, d. h. Städte plus ASN (in der Regel Internetdienstanbieter). Die Karte zeigt eine Kombination aus Verfügbarkeits- und Leistungsproblemen, die sich in letzter Zeit auf das Interneterlebnis von Kunden mit den wichtigsten Standorten und Diensten weltweit ausgewirkt haben. AWS

Woher stammen die Daten für die Karte?

Die Daten basieren auf einer Kombination aus aktivem und passivem Surfen im Internet. Weitere Informationen darüber, wie Internet Monitor Daten misst, finden Sie im Abschnitt So AWS werden Verbindungsprobleme gemessen.

Wie oft wird die Karte aktualisiert?

Die Internet-Wetterkarte wird alle 15 Minuten aktualisiert.

Welche Netzwerke werden auf Ausfälle überwacht?

AWS verfolgt Netzwerke auf der ganzen Welt, die wichtige IP-Präfixe darstellen, mit denen Kunden Internetverbindungen herstellen. AWS Wir untersuchen Ausfälle an den Standorten unserer Kunden, die für das Volumen des an das Netzwerk gesendeten und vom Netzwerk empfangenen Datenverkehrs am stärksten verantwortlich sind. AWS

Wodurch wird bestimmt, ob ein Internetereignis auf der Karte enthalten ist?

Hier sind einige allgemeine Kriterien, anhand derer wir bestimmen, ob ein Internet-Ereignis auf der Internet-Wetterkarte enthalten ist:

- AWS erkennt, dass ein Verfügbarkeits- oder Leistungsereignis vorliegt.
- Wenn das Ereignis nur von kurzer Dauer ist, z. B. weniger als 5 Minuten dauert, ignorieren wir es.
- Findet die Veranstaltung dann an einem Kundenstandort statt, der als Top-Talker eingestuft ist, gilt dies als Ausfall.

Welche Schwellenwerte werden für die Internet-Wetterkarte verwendet?

Die Schwellenwerte für die Bestimmung von Ausfällen sind für die Internet-Wetterkarte nicht statisch. Internet Monitor bestimmt anhand der Erkennung einer Abweichung von den erwarteten Werten, was ein Ereignis darstellt. Sie können mehr darüber erfahren, wie das funktioniert, indem Sie nachlesen, wie Internet Monitor bestimmt, wann Integritätsereignisse für Monitore erstellt werden müssen, die Sie mit dem Dienst erstellen. Wenn Sie einen Monitor erstellen, generiert Internet Monitor Integritätsmessungen für den Internetdatenverkehr, die für Ihren eigenen Anwendungsdatenverkehr spezifisch sind. Internet Monitor informiert Sie auch über Integritätsereignisse bei Problemen, die sich auf den Internetverkehr Ihrer Anwendung auswirken.

Was kann ich mit diesen Daten machen?

Die Internet-Wetterkarte bietet eine kurze Zusammenfassung der wichtigsten Internetereignisse, die sich in den letzten 24 Stunden auf der ganzen Welt ereignet haben. Sie hilft Ihnen, sich einen Eindruck von der Internetüberwachung zu verschaffen, ohne Ihren eigenen Internetverkehr in Internet Monitor einbinden zu müssen. Um das volle Potenzial der Internetüberwachungsfunktionen von Internet zu nutzen AWS und sie für Ihre Anwendungen und Dienste AWS, auf denen Sie gehostet werden, zu personalisieren, können Sie in Internet Monitor einen Monitor erstellen.

Wenn Sie einen Monitor erstellen, aktivieren Sie Internet Monitor, um die spezifischen Internetpfade zu identifizieren, die sich auf Ihre Anwendungsclients auswirken, und Sie

erhalten Zugriff auf Funktionen und Fähigkeiten, mit denen Sie Ihr Client-Erlebnis verbessern können. Außerdem werden Sie proaktiv über neue Internetprobleme informiert, die sich speziell auf Ihren Anwendungsdatenverkehr und Ihre Clients auswirken.

Wie kann ich weitere Informationen zu Veranstaltungen erhalten?

Klicken Sie auf der Karte auf einen Ausfall, um Details zu sehen, darunter den Beginn und das Ende einer Veranstaltung, die betroffene Stadt und die Vorabmitteilung sowie die Art des Problems (d. h. ein Leistungs- oder Verfügbarkeitsproblem).

Um detailliertere Informationen zu Ereignissen und benutzerdefinierte Messungen für Ihren Anwendungsdatenverkehr zu erhalten, erstellen Sie in Internet Monitor einen Monitor.

Beispielanwendungsfälle für Amazon CloudWatch Internet Monitor

In diesem Abschnitt beschreiben wir mehrere konkrete Beispiele mit Links zu Blogbeiträgen mit weiteren Details. Diese Beispiele zeigen, wie Sie die Funktionen von Amazon CloudWatch Internet Monitor nutzen können, um Ihre Anwendung zu überwachen und die Benutzererfahrung zu verbessern.

Richten Sie Benachrichtigungen ein und entscheiden Sie, welche Maßnahmen ergriffen werden sollen

Sie können Internet Monitor verwenden, um Einblicke in den Verlauf der durchschnittlichen Internet-Leistungsmetriken und in Zustandsereignisse einzelner Stadtnetze (Kundenstandort und ASN, in der Regel ein Internetdienstanbieter) zu erhalten. Mithilfe von Internet Monitor können Sie die Ereignisse identifizieren, die sich auf die Endbenutzererfahrung von Anwendungen auswirken, die auf Amazon Virtual Private Clouds (VPCs), Network Load Balancers WorkSpaces, Amazon oder Amazon gehostet werden. CloudFront

Nachdem Sie einen Monitor erstellt haben, haben Sie mehrere Möglichkeiten, wie Sie über Zustandsereignisse von Internet Monitor benachrichtigt werden können. Dazu gehören Benachrichtigungen, die auf CloudWatch Alarmen basieren und anhand von Ereignismetriken oder EventBridge Amazon-Regeln nach Gesundheitsereignissen filtern. Sie können verschiedene Optionen für Benachrichtigungen oder Aktionen wählen, die auf Alarmen basieren, darunter beispielsweise AWS SMS Benachrichtigungen oder Aktualisierungen einer CloudWatch Protokollgruppe.

Ein Beispiel mit ausführlicher Anleitung finden Sie im folgenden Blogbeitrag: Einführung in Amazon CloudWatch Internet Monitor.

Identifizieren Sie Latenzprobleme und verbessern Sie TTFB, um das Multiplayer-Spielerlebnis zu verbessern

Verwenden Sie Internet Monitor, um schnell zu erkennen, wo Spieler in globalen Cloud-Gaming-Apps weltweit Latenzprobleme haben, und erhalten Sie Einblicke zur Verbesserung der Leistung. Wenn Sie herausfinden, wo die Spieler derzeit die langsamste Time to First Byte (TTFB) haben, wissen Sie, wie Sie die Latenz verbessern können, um Ihre größte Spielerbasis zufriedener zu machen.

Wenn Sie nun bereit sind, den nächsten EC2-Server für Ihr Spiel bereitzustellen, wählen Sie den aus AWS-Region , der laut Internet Monitor den TTFB in dem Bereich mit der hohen Latenz und der großen Spielergruppe senkt.

Einzelheiten zur Einrichtung und Verwendung von Internet Monitor für diesen Anwendungsfall finden Sie im folgenden Blogbeitrag: Amazon CloudWatch Internet Monitor for a Better Gaming Experience verwenden.

Kontoübergreifende Beobachtbarkeit von Internet Monitor

Mit der kontenübergreifenden Beobachtbarkeit von Internet Monitor können Sie Ihre Anwendungen überwachen, die sich über mehrere AWS Konten innerhalb eines einzigen Kontos erstrecken. AWS-Region

Sie können Amazon CloudWatch Observability Access Manager verwenden, um eines oder mehrere Ihrer AWS Konten als Überwachungskonto einzurichten. Sie geben dem Überwachungskonto die Möglichkeit, Daten in Ihrem Quellkonto einzusehen, indem Sie eine Senke in Ihrem Überwachungskonto einrichten. Eine Senke ist eine Ressource, die einen Zuordnungspunkt in einem Überwachungskonto darstellt. Bei Internet Monitor ist der Ressourcenzuordnungspunkt ein Monitor. Sie verwenden die Sink, um eine Verbindung von Ihrem Quellkonto zu Ihrem Überwachungskonto herzustellen. Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.

Erforderliche -Ressourcen

Damit die kontenübergreifende Observability von CloudWatch Application Insights ordnungsgemäß funktioniert, stellen Sie sicher, dass die folgenden Telemetriearten über den CloudWatch Observability Access Manager gemeinsam genutzt werden.

- Monitore im Internetmonitor

- Metriken bei Amazon CloudWatch
- Gruppen in Amazon CloudWatch Logs protokollieren

Erste Schritte mit Amazon CloudWatch Internet Monitor mithilfe der Konsole

Um mit Amazon CloudWatch Internet Monitor zu beginnen, müssen Sie in Internet Monitor einen Monitor für Ihre Anwendung erstellen, indem Sie die verwendeten AWS Ressourcen hinzufügen und verschiedene Konfigurationsoptionen festlegen. Dieses Kapitel beschreibt das Verfahren zum Hinzufügen eines Monitors in der Konsole. Es enthält außerdem einen Abschnitt mit weiteren Informationen zu den Ressourcen in Internet Monitor sowie weitere Abschnitte mit Beschreibungen und Einschränkungen der verschiedenen Optionen, die Sie für Ihren Monitor konfigurieren können oder müssen.

Inhalt

Erstellen eines Monitors in Amazon CloudWatch Internet Monitor mithilfe der Konsole

Sie erstellen in Amazon CloudWatch Internet Monitor einen Monitor für Ihre Anwendung, indem Sie AWS Ressourcen hinzufügen, die sie verwendet, und dann mehrere Konfigurationsoptionen festlegen. Die Ressourcen, die Sie hinzufügen, Amazon Virtual Private Clouds (VPCs), Network Load Balancers (NLBs), CloudFront Distributionen oder WorkSpaces Verzeichnisse, stellen die Informationen für Internet Monitor bereit, um Internetverkehrsinformationen für Ihre Anwendung zuzuordnen. Warten Sie nach der Erstellung Ihres Monitors 15 bis 30 Minuten, um das spezifische Verkehrsprofil für den Einsatzort Ihrer Anwendung zu generieren. Anschließend können Sie

den Internet Monitor-Monitor oder andere Tools verwenden, um die Leistung und Verfügbarkeit Ihrer Client-Nutzung zu visualisieren und zu untersuchen. Diese Tools bieten Ihnen anhand der Messungen Ihres Anwendungsdatenverkehrs, die vom Monitor erfasst und veröffentlicht werden, z. B. in CloudWatch Logs.

Normalerweise ist es am einfachsten, einen Monitor in Internet Monitor für eine Anwendung zu erstellen. Innerhalb desselben Monitors können Sie die Messungen und Metriken in den Protokolldateien von Internet Monitor nach verschiedenen Standorten und ASNs (in der Regel Internetdienstanbieter) oder anderen Informationen durchsuchen und sortieren. Es ist beispielsweise nicht erforderlich, separate Monitore für Anwendungen in verschiedenen Bereichen zu erstellen.

Die folgenden Schritte führen Sie durch die Einrichtung Ihres Monitors mithilfe der Konsole. Beispiele für die Verwendung von Aktionen AWS Command Line Interface mit der Internet Monitor-API, zum Erstellen eines Monitors, zum Anzeigen von Ereignissen usw. finden Sie unterBeispiele für die Verwendung der CLI mit Amazon CloudWatch Internet Monitor.

So erstellen Sie einen Monitor mithilfe der Konsole

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im linken Navigationsbereich unter Netzwerküberwachung die Option Internetmonitor aus.
3. Klicken Sie auf Create monitor (Überwachung erstellen).
4. Geben Sie für Monitor name (Name des Monitors) den Namen ein, den Sie für diesen Monitor in Internet Monitor verwenden möchten.
5. Wählen Sie Add resources (Ressourcen hinzufügen) und wählen Sie dann die Ressourcen aus, um die Überwachungsgrenzen festzulegen, die Internet Monitor für diesen Monitor verwenden soll.

> **Note**
>
> Achten Sie auf Folgendes:
>
> - Um mit Internet Monitor aussagekräftige Ergebnisse zu generieren, müssen die von Ihnen hinzugefügten VPCs über ein konfiguriertes Internet Gateway mit dem Internet verbunden sein.
> - Sie können eine Kombination aus VPCs und CloudFront Distributionen hinzufügen, oder Sie können WorkSpaces Verzeichnisse hinzufügen, oder Sie können Network

Load Balancer hinzufügen. Sie können Network Load Balancer oder WorkSpaces Verzeichnisse nicht zusammen mit anderen Ressourcentypen hinzufügen.

6. Wählen Sie einen Prozentsatz Ihres Internetverkehrs aus, den Sie überwachen möchten.
7. Geben Sie optional zusätzliche Optionen unter Erweiterte Einstellungen an.

   - Für Maximale Anzahl von Stadtnetzen können Sie ein Limit für die Anzahl der Stadtnetze (Standorte und ASNs oder Internetdienstanbieter) festlegen, für die Internet Monitor den Datenverkehr überwachen soll. Sie können dies jederzeit ändern, indem Sie Ihren Monitor bearbeiten. Siehe Auswahl einer Höchstgrenze für Städtenetze.

     Um auf die Standardwerte zurückzusetzen, geben Sie `500000` ein.

     Wenn Sie eine Höchstgrenze für Stadtnetze festlegen, wird eine Obergrenze für die Anzahl der Stadtnetze festgelegt, die Internet Monitor für Ihre Anwendung überwacht, unabhängig von dem Prozentsatz des Datenverkehrs, den Sie überwachen möchten.
   - Optional können Sie einen Amazon-S3-Bucket-Namen und ein benutzerdefiniertes Präfix angeben, um Internetmessungen für alle überwachten Stadtnetze in Amazon S3 zu veröffentlichen.

     Internet Monitor veröffentlicht alle fünf Minuten die 500 wichtigsten Internet-Messwerte (nach Verkehrsaufkommen) für Ihre Anwendung in CloudWatch Logs. Wenn Sie sich dafür entscheiden, Messungen in S3 zu veröffentlichen, werden die Messungen trotzdem in CloudWatch Logs veröffentlicht. Weitere Informationen finden Sie unter Internet-Messungen in Amazon S3 in Amazon CloudWatch Internet Monitor veröffentlichen.
   - Optional können Sie ein Tag für Ihren Monitor hinzufügen.
8. Klicken Sie auf Create monitor (Überwachung erstellen).

Nachdem Sie einen Monitor erstellt haben, können Sie ihn jederzeit bearbeiten, z. B. um den Prozentsatz des Anwendungsverkehrs zu ändern, die maximale Anzahl der Stadtnetze zu aktualisieren oder Ressourcen hinzuzufügen oder zu entfernen. Sie können den Monitor auch löschen. Um diese Aufgaben auszuführen, wählen Sie in der Internet-Monitor-Konsole einen Monitor und dann eine Option im Menü Aktion aus. Beachten Sie, dass Sie den Namen eines Monitors nicht mehr ändern können.

So zeigen Sie das Internet-Monitor-Dashboard an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Netzwerküberwachung und dann Internetmonitor aus.

   Auf der Registerkarte Monitors (Monitore) wird die Liste von Monitoren angezeigt, die Sie erstellt haben.

Wählen Sie einen einzelnen Monitor aus, um weitere Informationen zu einem Monitor zu erhalten.

Hinzufügen von Ressourcen zu Ihrem Monitor

Wenn Sie einen Monitor erstellen, müssen Sie ihm die Ressourcen Ihrer Anwendung zuordnen: Amazon Virtual Private Clouds (VPCs), Network Load Balancers, CloudFront Amazon-Distributionen, Network Load Balancers (NLBs) oder Amazon-Verzeichnisse. WorkSpaces Dann weiß Internet Monitor, wo sich der mit dem Internet verbundene Datenverkehr und die Clients Ihrer Anwendung befinden, und kann ein Verkehrsprofil erstellen und verwalten, das die relevanten Messwerte festlegt, die für Ihren Monitor veröffentlicht werden sollen.

Sie können einem Monitor in Internet Monitor die folgenden Ressourcentypen als „überwachte Ressourcen“ hinzufügen. Beachten Sie, dass Internet Monitor das Zusammenfügen verschiedener Ressourcentypen in einem Monitor nicht unterstützt.

- VPCs: Jede VPC, die Sie einer Region hinzufügen, ist eine überwachte Ressource. Wenn Sie eine VPC hinzufügen, überwacht Internet Monitor den Datenverkehr für jede mit dem Internet verbundene Anwendung in der VPC, z. B. eine Anwendung, die auf einer Amazon EC2 EC2-Instance, hinter einem Network Load Balancer oder in einem Container gehostet wird. AWS Fargate
- Network Load Balancer: Jeder Network Load Balancer, den Sie hinzufügen, ist eine überwachte Ressource.
- CloudFront Verteilungen: Jede CloudFront Distribution, die Sie hinzufügen, ist eine überwachte Ressource.
- WorkSpaces Verzeichnisse: Jedes WorkSpaces Verzeichnis, das Sie einer Region hinzufügen, ist eine überwachte Ressource.

Wenn Sie den Datenverkehr für VPCs überwachen, wird der Datenverkehr für Anwendungen überwacht, die auf Load Balancern hinter der VPC gehostet werden. Sie können sich dafür

entscheiden, den Datenverkehr für einzelne Network Load Balancer zu überwachen, anstatt eine VPC mit mehreren Load Balancern zu überwachen. Dies kann zum Beispiel hilfreich sein, wenn Sie Features für eine bessere Leistung oder Effizienz auf der Ebene des Load Balancer verstehen und konfigurieren müssen. Oder Sie benötigen Compliance-Informationen auf der Ebene des Network Load Balancers.

Wenn Sie in Internet Monitor Ressourcen zu einem Monitor hinzufügen, sollten Sie Folgendes beachten:

- Um mit Internet Monitor aussagekräftige Ergebnisse zu generieren, müssen die von Ihnen hinzugefügten VPCs über ein konfiguriertes Internet Gateway mit dem Internet verbunden sein.
- Internet Monitor unterstützt nicht das Zusammenfügen verschiedener Ressourcentypen in einem Monitor.

Es gibt regionale Unterschiede bei den Opt-in-Regionen, die Sie berücksichtigen sollten, wenn Sie VPCs oder NLBs als Ressourcen hinzufügen. Weitere Informationen finden Sie unter Unterstützt AWS-Regionen für Amazon CloudWatch Internet Monitor.

Darüber hinaus gibt es Unterschiede bei den Ressourcen zur Messung der Latenzzeit auf der letzten Meile. Bei Latenzmessungen von Internet Monitor berücksichtigen VPCs, NLBs und WorkSpaces Verzeichnisse keine Latenz auf der letzten Meile.

Auswählen eines Prozentsatzes des zu überwachenden Anwendungsverkehrs

Die Abdeckung, die Sie für den Prozentsatz des zu überwachenden Anwendungsverkehrs wählen, bestimmt, wie viele Stadtnetze (Kundenstandorte und ASNs, in der Regel Internetdienstanbieter) für Ihre Anwendung überwacht werden, bis zu einer optionalen Höchstgrenze für Stadtnetze, die Sie ebenfalls festlegen können.

Wenn Sie sich dafür entscheiden, weniger als 100 % Ihres Anwendungsverkehrs zu überwachen, könnte eine Lücke in der Beobachtbarkeit Ihres Monitors bestehen. Das liegt daran, dass Sie sich dieser Probleme nicht bewusst sind, wenn Amazon CloudWatch Internet Monitor Gesundheitsereignisse verursacht, bei denen Sie den Datenverkehr nicht überwachen. Möglicherweise haben Sie auch eine geringere Abdeckung für die Leistungs- und Verfügbarkeitswerte der Informationen über den Client-Zugriff auf Ihre Anwendung.

In den folgenden Abschnitten werden Optionen beschrieben, mit denen Sie die prozentualen Einstellungen für Datenverkehr und Abdeckung untersuchen und sich ein Bild von den Auswirkungen einer Erhöhung oder Verringerung der Abdeckung machen können.

- Untersuchen Sie, wie Sie den Prozentsatz Ihres Anwendungsdatenverkehrs ändern können
- Anzahl der zu verschiedenen Prozentsätzen und Datenverkehrseinstellungen überwachten Stadtnetze anzeigen

Untersuchen Sie, wie Sie den Prozentsatz Ihres Anwendungsdatenverkehrs ändern können

Sie können die möglichen Werte erkunden, die Sie für die Änderung des Prozentsatzes für den Anwendungsverkehr verwenden möchten, indem Sie die Anzahl der überwachten Stadtnetze anzeigen, während Sie den Prozentsatz ändern. Das Verfahren in diesem Abschnitt enthält step-by-step Informationen.

In der Konsole von Internet Monitor können Sie versuchen, den Prozentsatz des Anwendungsdatenverkehrs für Ihren Monitor zu erhöhen oder zu verringern, und die geschätzte Anzahl Ihrer Stadtnetze anzeigen, die dadurch abgedeckt würden. Mit dieser Option können Sie schnell sehen, wie sich eine Änderung Ihres Datenverkehrsanteils auf die Anzahl der überwachten Stadtmonitore auswirkt. Dies kann Ihnen helfen, ein Gefühl dafür zu entwickeln, wie hoch der Prozentsatz des Datenverkehrs für Ihre Anwendung sein sollte.

Untersuchung der Überwachungsabdeckung durch Erhöhen und Verringern des Prozentsatzes des Anwendungsverkehrs

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im linken Navigationsbereich unter Netzwerküberwachung die Option Internetmonitor aus.
3. Wählen Sie in Ihrer Monitorliste einen Monitor aus.
4. Wählen Sie auf der Registerkarte Übersicht im Abschnitt Überwachter Datenverkehr das Diagramm mit dem Prozentsatz und wählen Sie dann Überwachungsabdeckung aktualisieren.
5. Klicken Sie im Dialogfeld Umfang der Überwachungsabdeckung untersuchen und festlegen auf die Pfeile, um den Prozentsatz des zu überwachenden Datenverkehrs zu erhöhen oder zu verringern. Wenn Sie die Option 100 % Datenverkehr wählen, können Sie sehen, wie viele Stadtnetze mit voller Abdeckung für die Überwachung Ihrer Anwendung überwacht werden.
6. Um mehr darüber zu erfahren, wie sich die Anzahl der überwachten Stadtnetzwerke (hier geschätzt) auf Ihre Kosten auswirken könnte, klicken Sie auf den Link zum CloudWatch Preisrechner und scrollen Sie dann nach unten zu Internet Monitor.

7. Um einen neuen Prozentsatz des zu überwachenden Datenverkehrs festzulegen, wählen Sie Überwachungsabdeckung aktualisieren. Oder wählen Sie Abbrechen, um die aktuelle Abdeckung beizubehalten.

Anzahl der zu verschiedenen Prozentsätzen und Datenverkehrseinstellungen überwachten Stadtnetze anzeigen

Sie können die Anzahl der Stadtnetze anzeigen, die für Ihre Anwendung mit verschiedenen Prozentsätzen des Anwendungsverkehrs überwacht werden. Das Verfahren in diesem Abschnitt enthält step-by-step Informationen.

In der Konsole von Internet Monitor können Sie Diagramme anzeigen, die zeigen, wie sich die Abdeckung Ihrer Stadtnetze bei unterschiedlichen Anwendungsverkehrsanteilen in einem von Ihnen festgelegten Zeitintervall ändern würde. Auf diese Weise können Sie die Überwachungsabdeckung für Ihre Anwendung bei einem bestimmten Prozentsatz des Datenverkehrs schnell in einem einzigen Diagramm anzeigen und vergleichen.

So zeigen Sie Diagramme zum prozentualen Anteil des Anwendungsverkehrs und der entsprechenden Abdeckung der Stadtnetze an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im linken Navigationsbereich unter Netzwerküberwachung die Option Internetmonitor aus.
3. Wählen Sie in Ihrer Monitorliste einen Monitor aus.
4. Wählen Sie die Registerkarte Einblicke in den Datenverkehr, und scrollen Sie nach unten zu den Diagrammen zum Internetverkehr.
5. Wählen Sie unter Optionen für die Verkehrsabdeckung vergleichen in der Dropdown-Liste einen oder mehrere Prozentsätze aus. Sie können einen oder mehrere Prozentsätze für den Anwendungsverkehr auswählen. Das Diagramm mit der Gesamtzahl der überwachten Stadtnetze wird dann aktualisiert, um die Überwachungsabdeckung anzuzeigen, die Internet Monitor für diesen Prozentsatz bietet. Wenn Sie die Option Stadtnetze mit 100 % Datenverkehr wählen, können Sie sehen, wie viele Stadtnetze mit voller Abdeckung für die Überwachung Ihrer Anwendung überwacht werden.

Beachten Sie Folgendes:

- Die Abdeckung des Datenverkehrs wird auf der Grundlage der Anzahl der Stadtnetze berechnet, die in der vorangegangenen Stunde Ihres Anwendungsverkehrs genutzt wurden. Das bedeutet, dass nach der Auswahl eines bestimmten Prozentsatzes des zu überwachenden Datenverkehrs möglicherweise weniger Stadtnetze für Ihre Anwendung überwacht werden, als hier in der Grafik zum Vergleich der Verkehrsabdeckung angezeigt wird.
- Um sicherzustellen, dass Ihr gesamter Anwendungsdatenverkehr überwacht wird, setzen Sie `TrafficPercentageToMonitor` auf 100 und stellen `MaxCityNetworksToMonitor` nicht ein. Alternativ können Sie `MaxCityNetworksToMonitor` auch auf 500 000 setzen, die Obergrenze in Internet Monitor.
- Wenn Sie eine Höchstgrenze für Stadtnetze festlegen, übersteigt die Gesamtzahl der überwachten Stadtnetze niemals diese Grenze, unabhängig von der von Ihnen gewählten Option für den prozentualen Anteil des Anwendungsverkehrs.
- Sie können mehr darüber erfahren, wie sich die Anzahl der überwachten Stadtnetze auf Ihre Kosten auswirken kann. Scrollen Sie auf der CloudWatch Seite Preisrechner für nach unten zu Internet Monitor.

Um einen neuen Prozentsatz des zu überwachenden Datenverkehrs festzulegen, wählen Sie unter Andere Verkehrsabdeckungsoptionen erkunden die Option Überwachungsabdeckung aktualisieren. Wählen Sie im Dialogfeld einen Prozentsatz des Datenverkehrs aus und wählen Sie dann Überwachungsabdeckung aktualisieren.

Auswahl einer Höchstgrenze für Städtenetze

Amazon CloudWatch Internet Monitor kann Ihren Anwendungsverkehr für einige oder alle Standorte überwachen, an denen Kunden auf Ihre Anwendungsressourcen zugreifen, sowie für alle ASNs (in der Regel Internetdienstanbieter), über die sie auf Ihre Anwendung zugreifen, d. h. die Stadtnetzwerke für Ihren Anwendungs-Internetverkehr. Sie wählen bei der Erstellung Ihres Monitors einen Prozentsatz des zu überwachenden Anwendungsverkehrs aus, den Sie jederzeit durch Bearbeiten des Monitors aktualisieren können.

Zusätzlich zur Festlegung eines Prozentsatzes für den Datenverkehr können Sie auch eine Höchstgrenze für die Anzahl der überwachten Stadtnetze festlegen. In diesem Abschnitt wird beschrieben, wie das Stadtnetz-Limit Ihnen bei der Verwaltung der Abrechnungskosten helfen kann, und Sie erhalten Informationen und ein Beispiel, das Ihnen bei der Festlegung eines Limits hilft.

Die Höchstgrenze, die Sie für die Anzahl der Stadtnetze festlegen, trägt dazu bei, Ihre Rechnung vorhersehbar zu gestalten. Weitere Informationen finden Sie unter CloudWatch Amazon-Preise.

Mithilfe des CloudWatch Preisrechners können Sie auch herausfinden, wie sich unterschiedliche Werte für die Anzahl der tatsächlich überwachten Stadtnetze auf Ihre Rechnung auswirken können. Scrollen Sie auf der CloudWatch Seite Preisrechner für nach unten zu Internet Monitor, um weitere Optionen zu finden.

Informationen zur Aktualisierung Ihres Monitors und zur Änderung des maximalen Limits für Stadtnetze finden Sie unter Einen Monitor von Internet Monitor bearbeiten oder löschen.

Wie die Abrechnung mit Höchstgrenzen für Stadtnetze funktioniert

Die Festlegung einer Höchstgrenze für die Anzahl der überwachten Stadtnetze kann dazu beitragen, unerwartete Kosten in Ihrer Rechnung zu vermeiden. Dies ist beispielsweise nützlich, wenn Ihre Verkehrsmuster stark variieren. Die Fakturierungskosten erhöhen sich für jedes Stadtnetz, das nach den ersten enthaltenen 100 Stadtnetzen überwacht wird (über alle Monitore pro Konto). Wenn Sie eine Höchstgrenze für Stadtnetze festlegen, wird die Anzahl der Stadtnetze, die Internet Monitor für Ihre Anwendung überwacht, begrenzt, unabhängig von dem Prozentsatz des Datenverkehrs, den Sie überwachen möchten.

Sie zahlen nur für die Anzahl der tatsächlich überwachten Stadtnetze. Mit der von Ihnen gewählten Höchstgrenze für das Stadtnetz können Sie eine Obergrenze für die Gesamtmenge festlegen, die bei der Überwachung des Datenverkehrs durch Internet Monitor berücksichtigt werden kann. Sie können die Höchstgrenze jederzeit ändern, indem Sie Ihren Monitor bearbeiten.

Um die Optionen zu erkunden, scrollen Sie auf der CloudWatch Seite Preisrechner für Seite nach unten zu Internet Monitor. Weitere Informationen zu den Preisen von Internet Monitor finden Sie im Abschnitt Internet Monitor auf der CloudWatch Amazon-Preisseite.

Wie man eine Höchstgrenze für Stadtnetze auswählt

Um zu entscheiden, welche Höchstgrenze Sie für Stadtnetze wählen sollten, überlegen Sie, wie viel Datenverkehr Sie für Ihre Anwendung überwachen möchten. Die folgenden Internet-Monitor-Metriken können Ihnen helfen, Ihre Verkehrsnutzung und -abdeckung zu analysieren, nachdem Sie Ihren Monitor erstellt haben: `CityNetworksMonitored`, `TrafficMonitoredPercent` und eine oder mehrere der `CityNetworksFor`*`NN`*`PercentTraffic`-Metriken, wobei *`NN`* ein Prozentwert ist, der einer der folgenden Werte ist: 25, 50, 90, 95, 99, oder 100. Weitere Definitionen für diese und alle anderen Internet-Monitor-Metriken finden Sie unter Verwenden von CloudWatch Metriken mit Amazon CloudWatch Internet Monitor.

Um ein Übersichtsdiagramm Ihrer Internet-Traffic-Abdeckung zu sehen, rufen Sie im CloudWatch Dashboard den Tab Traffic Insights auf und wählen Sie im Bereich Internet-Traffic-Grafiken eine

Option für Vergleichsoptionen zur Datenverkehrsabdeckung aus. Das in diesem Abschnitt gezeigte Diagramm zeigt die tatsächliche Anzahl der Stadtnetze, die für Ihre Anwendung überwacht werden, sowie die Diagrammlinien für verschiedene Prozentsätze des Anwendungsverkehrs, die Sie in der Dropdown-Liste auswählen. Weitere Informationen finden Sie unter Den Prozentsatz Ihres Anwendungsverkehrs festlegen.

Um Ihre Optionen genauer zu untersuchen, können Sie die Internet-Monitor-Metriken verwenden, wie in den folgenden Beispielen beschrieben. Diese Beispiele zeigen Ihnen, wie Sie die für Sie am besten geeignete Höchstgrenze für Stadtnetze auswählen können, je nachdem, wie weit Sie den Internetverkehr Ihrer Anwendung abdecken möchten. Mithilfe der Abfragen für Internet Monitor-Metriken in CloudWatch Metriken können Sie mehr über die Abdeckung des Internetverkehrs in Ihrer Anwendung erfahren.

Beispiel für die Bestimmung einer Höchstgrenze für Stadtnetze

Nehmen wir an, Sie haben eine Überwachungsobergrenze von 100 Stadtnetzen festgelegt und auf Ihre Anwendung wird von Kunden in 2 637 Stadtnetzen zugegriffen. Unter CloudWatch Metriken würden die folgenden Internet Monitor-Metriken zurückgegeben:

```
CityNetworksMonitored 100
TrafficMonitoredPercent  12.5
CityNetworksFor90PercentTraffic  2143
CityNetworksFor100PercentTraffic  2637
```

Aus diesem Beispiel können Sie ersehen, dass Sie derzeit 12,5 % Ihres Internetverkehrs überwachen, wobei die Höchstgrenze auf 100 Stadtnetze festgelegt ist. Wenn Sie 90 % Ihres Datenverkehrs überwachen möchten, gibt Ihnen die nächste Metrik Auskunft darüber: `CityNetworksFor90PercentTraffic` zeigt an, dass Sie für eine 90%ige Abdeckung 2 143 Stadtnetze überwachen müssten. Aktualisieren Sie dazu Ihren Monitor und setzen Sie die maximale Anzahl der Stadtnetze auf 2 143.

Nehmen wir an, Sie möchten eine 100%ige Überwachung des Internetverkehrs für Ihre Anwendung. Die nächste Metrik, `CityNetworksFor100PercentTraffic`, zeigt an, dass Sie dazu Ihren Monitor aktualisieren müssen, um die maximale Anzahl der Stadtnetze auf 2 637 zu setzen.

Wenn Sie nun das Maximum auf 5 000 Stadtnetze setzen, sehen Sie die folgenden Metriken, da dies mehr als 2 637 sind:

```
CityNetworksMonitored 2637
```

```
TrafficMonitoredPercent  100
CityNetworksFor90PercentTraffic  2143
CityNetworksFor100PercentTraffic  2637
```

Anhand dieser Metriken können Sie sehen, dass Sie mit dem höheren Limit alle 2 637 Stadtnetze überwachen, was 100 % Ihres Internetverkehrs entspricht.

Internet-Messungen in Amazon S3 in Amazon CloudWatch Internet Monitor veröffentlichen

Sie können festlegen, dass Amazon CloudWatch Internet Monitor Internetmessungen für Ihren mit dem Internet verbundenen Datenverkehr zu den überwachten Stadtnetzen (Kundenstandorte und ASNs, in der Regel Internetdienstanbieter) in Ihrem Monitor veröffentlicht, und zwar bis zum Servicelimit von 500.000 Stadtnetzwerken. Internet Monitor veröffentlicht automatisch alle fünf Minuten Internet-Messungen in CloudWatch Logs für die 500 größten Stadtnetzwerke (nach Verkehrsaufkommen) für jeden Monitor. Zu den Messungen, die in S3 veröffentlicht werden, gehören auch die 500 wichtigsten, die in CloudWatch Logs veröffentlicht wurden.

Sie können die Option zur Veröffentlichung in S3 wählen und den Bucket angeben, in dem die Messungen veröffentlicht werden sollen, wenn Sie Ihren Monitor erstellen oder aktualisieren. Der Bucket muss bereits in S3 erstellt worden sein, bevor Sie ihn in Internet Monitor angeben können. Es gibt ein Service-Limit von 500 000 Stadtnetzen für Internetmessungen, die in S3 veröffentlicht werden. Internet Monitor veröffentlicht Internetmessungen in S3 als Ereignisse, einer Reihe von komprimierten Protokolldatei-Objekten, die in dem Bucket gespeichert werden.

Wenn Sie den S3-Bucket für Internet Monitor erstellen, in dem Messungen veröffentlicht werden sollen, stellen Sie sicher, dass Sie die in CloudWatch Logs angegebenen Hinweise zu Berechtigungen befolgen. Dadurch wird sichergestellt, dass Internet Monitor Protokolle direkt in S3 veröffentlichen AWS kann und dass bei Bedarf die Ressourcenrichtlinien für die Protokollgruppe, die die Protokolle empfängt, erstellt und geändert werden können. Weitere Informationen finden Sie unter An Logs sendete CloudWatch Logs im Amazon CloudWatch Logs-Benutzerhandbuch.

Die veröffentlichten Protokolldateien sind komprimiert. Wenn Sie die Protokolldateien mit der Amazon-S3-Konsole öffnen, werden sie dekomprimiert und die Ereignisse der Internetmessung werden angezeigt. Wenn Sie die Dateien herunterladen, müssen Sie sie dekomprimieren, um die Datensätze anzuzeigen.

Sie können die Internetmessungen in den Protokolldateien auch mit Amazon Athena abfragen. Amazon Athena ist ein interaktiver Abfrageservice, der die Analyse von Daten in Amazon S3 mit

Hilfe von Standard-SQL erleichtert. Weitere Informationen finden Sie unter Verwendung von Amazon Athena zur Abfrage von Internetmessungen in Amazon-S3-Protokolldateien.

Verwenden eines Internet-Monitor-Monitors

Es gibt mehrere Möglichkeiten, einen Amazon CloudWatch Internet Monitor-Monitor zu verwenden, nachdem Sie ihn erstellt haben: Sie können beispielsweise Informationen im CloudWatch Dashboard anzeigen, Informationen mithilfe von abrufen und Gesundheitswarnungen einrichten. AWS Command Line Interface

Ihr Monitor liefert Informationen über Ihre Anwendung und Ihre Konfigurationspräferenzen, so dass Internet Monitor Messungen und Metriken anpassen und in Ereignissen für Sie veröffentlichen kann. Internet Monitor sammelt Messungen der globalen Infrastruktur für AWS. Bei diesen Messungen handelt es sich um eine enorme Menge an Informationen zur Netzwerkleistung und -verfügbarkeit aus der ganzen Welt. Anhand der Informationen aus den Ressourcen, die Sie für Ihre Anwendung hinzufügen, veröffentlicht Internet Monitor für Sie Leistungs- und Verfügbarkeitsmessungen, die sich auf die Stadtnetze (d. h. Kundenstandorte und ASNs, in der Regel Internetdienstanbieter oder ISPs) beziehen, in denen Ihre Anwendung aktiv ist. Daher sind die Messungen und Messwerte im Internet Monitor-Dashboard und in den CloudWatch Protokollen — über Verfügbarkeit, Leistung, übertragene überwachte Byte und Round-Trip-Zeit — spezifisch für Ihre Kundenstandorte und ASNs.

Internet Monitor ermittelt auch, wann Anomalien in Bezug auf Leistung und Verfügbarkeit vorliegen. Standardmäßig überlagert Internet Monitor Ihren Datenverkehr mit den Verfügbarkeits- und Leistungsmessungen, die für jedes Quell- und Zielpaar an Ihren Kundenstandorten erfasst wurden, um festzustellen, wann es zu nennenswerten Leistungs- oder Verfügbarkeitseinbußen kommt. AWS Wenn es zu einer signifikanten Beeinträchtigung der Standorte und des Umfangs Ihrer Anwendung kommt, erzeugt Internet Monitor ein Zustandsereignis und veröffentlicht Informationen über das Problem in Ihrem Monitor.

Nachdem Sie einen Monitor erstellt haben, können Sie ihn verwenden, um auf die von Internet Monitor bereitgestellten Informationen zuzugreifen oder sich über diese informieren zu lassen:

- Verwenden Sie das CloudWatch Dashboard, um Leistungs-, Verfügbarkeits- und Integritätsereignisse anzuzeigen und zu untersuchen, die historischen Daten Ihrer Anwendung zu untersuchen und Einblicke in neue Möglichkeiten zur Konfiguration Ihrer Anwendung für eine bessere Leistung zu erhalten. Für weitere Informationen siehe:
  - Verfolgen von Leistung und Verfügbarkeit in Echtzeit in Amazon CloudWatch Internet Monitor (Registerkarte „Übersicht“)

  - Filtern und Anzeigen von historischen Daten in Amazon CloudWatch Internet Monitor (Registerkarte Historischer Explorer)
  - Erhalten von Erkenntnissen zur Verbesserung der Anwendungsleistung in Amazon CloudWatch Internet Monitor (Registerkarte Traffic Insights)
- Konfigurieren Sie Schwellenwerte für Zustandsereignisse, um zu ändern, was Internet Monitor dazu veranlasst, ein Zustandsereignis für Ihre Anwendung zu erzeugen. Sie können allgemeine Schwellenwerte und lokale (Stadtnetz) Schwellenwerte konfigurieren. Weitere Informationen finden Sie unter Schwellenwerte für Zustandsereignisse ändern.
- Verwenden Sie AWS CLI Befehle zusammen mit Internet Monitor-API-Aktionen, um Informationen zum Verkehrsprofil anzuzeigen, Messwerte einzusehen, Gesundheitsereignisse aufzulisten usw. Weitere Informationen hierzu finden Sie unter Beispiele für die Verwendung der CLI mit Amazon CloudWatch Internet Monitor.
- Verwenden Sie CloudWatch Standardtools wie CloudWatch Contributor Insights, CloudWatch Metrics Explorer und CloudWatch Logs Insights, um die Daten in CloudWatch zu visualisieren. Weitere Informationen hierzu finden Sie unter Erkunden Sie Ihre Daten mit CloudWatch Tools und der Internet Monitor-Abfrageschnittstelle.
- Verwenden Sie Athena mit S3-Protokollen, um auf die Internet-Monitor-Messungen für Ihre Anwendung zuzugreifen und diese zu analysieren, wenn Sie die Veröffentlichung von Messungen in S3 aktiviert haben.
- Erstellen Sie EventBridge Amazon-Benachrichtigungen, um Sie zu benachrichtigen, wenn Internet Monitor feststellt, dass ein Gesundheitsereignis vorliegt. Weitere Informationen hierzu finden Sie unter Amazon CloudWatch Internet Monitor mit Amazon verwenden EventBridge.
- Erhalten Sie automatisch eine AWS Health Dashboard Benachrichtigung, wenn Internet Monitor feststellt, dass ein Problem durch das AWS Netzwerk verursacht wird. Die Benachrichtigung enthält die Schritte, AWS mit denen das Problem behoben werden soll.

Einen Monitor von Internet Monitor bearbeiten oder löschenq

Über das Aktionsmenü können Sie einen Monitor in Amazon CloudWatch Internet Monitor bearbeiten oder löschen, nachdem Sie ihn erstellt haben. Beispielsweise können Sie einen Monitor so bearbeiten, dass er Folgendes tut:

- Den Prozentsatz des zu überwachenden Anwendungsverkehrs ändern
- Festlegen oder Aktualisieren der Höchstgrenze für Stadtnetze
- Schwellenwerte für Zustandsereignisse bei Verfügbarkeits- oder Leistungsbewertungen ändern

- Ressourcen hinzufügen oder entfernen
- Die Veröffentlichung von Ereignissen in Amazon S3 aktivieren oder aktualisieren

Sie können einen Monitor auch löschen. Beachten Sie, dass Sie den Namen eines Monitors nicht mehr ändern können, nachdem Sie ihn erstellt haben.

Um Änderungen an einem Monitor vorzunehmen oder einen Monitor zu löschen, verwenden Sie eines der folgenden Verfahren.

So bearbeiten Sie einen Monitor

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im linken Navigationsbereich unter Netzwerküberwachung die Option Internetmonitor aus.
3. Wählen Sie Ihren Monitor und dann das Menü Aktion.
4. Wählen Sie Monitor aktualisieren.
5. Nehmen Sie die gewünschten Aktualisierungen vor. Um beispielsweise den Prozentsatz des zu überwachenden Datenverkehrs zu ändern, wählen Sie unter Zu überwachender Anwendungsverkehr einen Prozentsatz aus oder geben ihn ein.
6. Wählen Sie Aktualisieren.

Um einen Monitor zu löschen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im linken Navigationsbereich unter Netzwerküberwachung die Option Internetmonitor aus.
3. Wählen Sie Ihren Monitor und dann das Menü Aktion.
4. Wählen Sie Disable (deaktivieren) aus.
5. Wählen Sie erneut das Aktionsmenü und dann Löschen aus.

Weitere Informationen zu den Optionen, die Sie aktualisieren können, finden Sie im Folgenden:

- Weitere Informationen zu Ressourcen, die Sie in Internet Monitor hinzufügen, finden Sie unter Hinzufügen von Ressourcen zu Ihrem Monitor.

- Weitere Informationen zum Prozentsatz des Anwendungsverkehrs finden Sie unter Auswählen eines Prozentsatzes des zu überwachenden Anwendungsverkehrs.
- Weitere Informationen zum Ändern des Schwellenwerts für Zustandsereignisse finden Sie unter Schwellenwerte für Zustandsereignisse ändern.
- Weitere Informationen zur Höchstgrenze für Stadtnetze finden Sie unter Auswahl einer Höchstgrenze für Städtenetze.
- Weitere Informationen darüber, wie Sie Ereignisse auf S3 veröffentlichen können, finden Sie unter Internet-Messungen in Amazon S3 in Amazon CloudWatch Internet Monitor veröffentlichen.

Einen Amazon CloudWatch Internet Monitor-Monitor mit Amazon VPC hinzufügen oder erstellen

Wenn Sie eine Amazon Virtual Private Cloud Cloud-VPC in der erstellen AWS Management Console, können Sie optional auch die Überwachung dafür in Amazon CloudWatch Internet Monitor einrichten. Sie können die VPC zu einem vorhandenen Monitor hinzufügen oder sich dafür entscheiden, einen neuen Monitor für die VPC in der Amazon-VPC-Konsole zu erstellen.

Verwenden Sie Internet Monitor mit Ihrer VPC, um Messungen und Metriken über Verfügbarkeit, Leistung, überwachte übertragene Bytes und Round-Trip-Zeiten anzuzeigen und auszuwerten, die für die Kundenstandorte und ASNs Ihrer Anwendung (in der Regel Internetdienstanbieter) spezifisch sind. Internet Monitor ermittelt auch, wann Anomalien in Bezug auf Leistung und Verfügbarkeit vorliegen, und erstellt Zustandsereignisse in Ihrem Monitor, über die Sie benachrichtigt werden können. Weitere Informationen darüber, wie Sie einen Monitor verwenden können, um die Benutzererfahrung Ihrer Kunden mit Ihrer Anwendung zu verwalten und zu verbessern, finden Sie unter Verwenden eines Internet-Monitor-Monitors.

> **Important**
>
> Um einen Monitor zu erstellen oder einem vorhandenen Monitor eine VPC hinzuzufügen, müssen Sie über die richtigen Berechtigungen verfügen. Weitere Informationen finden Sie unter Identity and Access Management für Amazon CloudWatch Internet Monitor.

Eine VPC zu einem vorhandenen Monitor hinzufügen

Sie können wählen, dass Amazon CloudWatch Internet Monitor eine neue VPC zu einem vorhandenen Monitor für Sie hinzufügt, wenn Sie die VPC in der erstellen. AWS Management

Console Warten Sie nach dem Hinzufügen der VPC einige Minuten. Dann werden Metriken für die VPC auf der Internet Monitor-Konsole angezeigt.

Sie können den Monitor jederzeit bearbeiten, um die VPC zu entfernen oder eine weitere VPC oder andere Ressourcen hinzuzufügen. Sie können auch den Prozentsatz des Datenverkehrs ändern, den Sie überwachen, oder andere Änderungen vornehmen. Wenn Sie sich dafür entscheiden, die VPC vom Monitor zu entfernen, wird der Datenverkehr von Kunden zu dieser VPC nicht mehr von Internet Monitor überwacht.

Weitere Informationen zum Aktualisieren eines Monitors finden Sie unter Einen Monitor von Internet Monitor bearbeiten oder löscheng.

Einen Monitor für eine VPC erstellen

Wenn Sie sich dafür entscheiden, einen Monitor für eine VPC zu erstellen, führt Sie der Assistent zum Erstellen eines Monitors durch die einzelnen Schritte. Sie fügen die VPC als überwachte Ressource hinzu, wenn Sie den Monitor erstellen. Wenn Sie möchten, können Sie auch einen Prozentsatz des Client-Datenverkehrs auswählen, den Sie für Ihre Anwendung überwachen möchten (die Standardeinstellung ist 100%).

Sie können mehr erfahren, indem Sie sich die Informationen in Erstellen eines Monitors in Amazon CloudWatch Internet Monitor mithilfe der Konsole ansehen.

Preisgestaltung

Mit Amazon CloudWatch Internet Monitor zahlen Sie nur für das, was Sie tatsächlich nutzen. Die Preisgestaltung für Internet Monitor besteht aus zwei Komponenten: einer Gebühr pro überwachter Ressource und einer Stadtnetzgebühr. Ein Stadtnetzwerk ist der Standort, von dem aus Kunden auf Ihre Anwendungsressourcen zugreifen, und das Netzwerk (eine ASN, z. B. ein Internetdienstanbieter oder ISP), über das Kunden auf die Ressourcen zugreifen.

Weitere Informationen, einschließlich Preisbeispielen, finden Sie unter Preise für Amazon CloudWatch Internet Monitor

Beenden der Überwachung einer VPC

Wenn Sie die Überwachung Ihrer VPC-Ressource mit Internet Monitor beenden möchten, gehen Sie in der Internet Monitor-Konsole wie folgt vor:

So entfernen Sie Ressourcen von einem Monitor

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

2. Wählen Sie im linken Navigationsbereich unter Netzwerküberwachung die Option Internetmonitor aus.
3. Wählen Sie Ihren Monitor und dann das Menü Aktion.
4. Wählen Sie Monitor aktualisieren.
5. Wählen Sie unter Ressourcen hinzugefügt die Option Ressourcen entfernen aus.
6. Wählen Sie die zu entfernende VPC und anschließend Entfernen aus.
7. Wählen Sie Aktualisieren.

Fügen Sie einen Amazon CloudWatch Internet Monitor-Monitor hinzu oder erstellen Sie ihn mit CloudFront

Im Metrik-Dashboard für eine Verteilung in der CloudFront Amazon-Konsole können Sie eine zusätzliche Überwachung für eine Verteilung in Amazon CloudWatch Internet Monitor einrichten. Sie können die Verteilung zu einem vorhandenen Monitor hinzufügen oder einen neuen Monitor für die Verteilung erstellen.

Wenn Sie Internet Monitor mit Ihrer CloudFront Distribution verwenden, können Sie Messungen und Messwerte zu Verfügbarkeit, Leistung, übertragenen überwachten Bytes und Round-Trip-Zeiten anzeigen und auswerten, die für die Client-Standorte und ASNs (in der Regel Internetdienstanbieter) Ihrer Anwendung spezifisch sind. Internet Monitor ermittelt auch, wann Anomalien in Bezug auf Leistung und Verfügbarkeit vorliegen, und erstellt Zustandsereignisse in Ihrem Monitor, über die Sie benachrichtigt werden können. Weitere Informationen darüber, wie Sie einen Monitor verwenden können, um die Benutzererfahrung Ihrer Kunden mit Ihrer Anwendung zu verwalten und zu verbessern, finden Sie unter Verwenden eines Internet-Monitor-Monitors.

> **Important**
>
> Um einen Monitor zu erstellen oder einem vorhandenen Monitor eine Distribution hinzuzufügen, müssen Sie über die richtigen Berechtigungen verfügen. Weitere Informationen finden Sie unter Identity and Access Management für Amazon CloudWatch Internet Monitor.

Fügen Sie einem vorhandenen Monitor eine Distribution hinzu

Sie können festlegen, dass Internet Monitor direkt über das CloudFront Metrik-Dashboard im eine Verteilung zu einem vorhandenen Monitor hinzufügt AWS Management Console. Nachdem Sie die

Verteilung hinzugefügt haben, warten Sie einige Minuten, bis die Metriken für die Verteilung auf der Internet Monitor-Konsole angezeigt werden.

Sie können den Monitor jederzeit bearbeiten, um die Verteilung zu entfernen oder eine weitere Distribution oder andere Ressourcen hinzuzufügen. Sie können auch den Prozentsatz des Datenverkehrs ändern, den Sie überwachen, oder andere Änderungen vornehmen. Wenn Sie sich dafür entscheiden, die Verteilung vom Monitor zu entfernen, wird der Datenverkehr von Clients zu dieser Verteilung nicht mehr von Internet Monitor überwacht.

Weitere Informationen zum Aktualisieren eines Monitors finden Sie unter Einen Monitor von Internet Monitor bearbeiten oder löscheng.

Erstellen Sie einen Monitor für eine Verteilung

Wenn Sie sich dafür entscheiden, einen Monitor für eine Distribution zu erstellen, führt Sie der Assistent zum Erstellen eines Monitors durch die einzelnen Schritte. Sie fügen die Verteilung als überwachte Ressource hinzu, wenn Sie den Monitor erstellen. Wenn Sie möchten, können Sie auch einen Prozentsatz des Client-Datenverkehrs auswählen, den Sie für Ihre Anwendung überwachen möchten (die Standardeinstellung ist 100%).

Sie können mehr erfahren, indem Sie sich die Informationen in Erstellen eines Monitors in Amazon CloudWatch Internet Monitor mithilfe der Konsole ansehen.

Preisgestaltung

Mit Amazon CloudWatch Internet Monitor zahlen Sie nur für das, was Sie tatsächlich nutzen. Die Preisgestaltung für Internet Monitor besteht aus zwei Komponenten: einer Gebühr pro überwachter Ressource und einer Stadtnetzgebühr. Ein Stadtnetzwerk ist der Standort, von dem aus Kunden auf Ihre Anwendungsressourcen zugreifen, und das Netzwerk (eine ASN, z. B. ein Internetdienstanbieter oder ISP), über das Kunden auf die Ressourcen zugreifen.

Weitere Informationen, einschließlich Preisbeispielen, finden Sie unter Preise für Amazon CloudWatch Internet Monitor

Beenden Sie die Überwachung einer Distribution

Wenn Sie die Überwachung Ihrer Vertriebsressource mit Internet Monitor beenden möchten, gehen Sie in der Internet Monitor-Konsole wie folgt vor:

So entfernen Sie Ressourcen von einem Monitor

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

2. Wählen Sie im linken Navigationsbereich unter Netzwerküberwachung die Option Internetmonitor aus.
3. Wählen Sie Ihren Monitor und dann das Menü Aktion.
4. Wählen Sie Monitor aktualisieren.
5. Wählen Sie unter Ressourcen hinzugefügt die Option Ressourcen entfernen aus.
6. Wählen Sie die zu entfernende Distribution aus, und klicken Sie dann auf Entfernen.
7. Wählen Sie Aktualisieren.

Beispiele für die Verwendung der CLI mit Amazon CloudWatch Internet Monitor

Dieser Abschnitt enthält Beispiele für die Verwendung von Vorgängen AWS Command Line Interface mit Amazon CloudWatch Internet Monitor.

Bevor Sie beginnen, stellen Sie sicher, dass Sie sich AWS CLI mit demselben AWS Konto anmelden, das die Amazon Virtual Private Clouds (VPCs), Network Load Balancers, CloudFront Amazon-Distributionen oder WorkSpaces Amazon-Verzeichnisse enthält, die Sie überwachen möchten. Internet Monitor unterstützt nicht den kontoübergreifenden Zugriff auf Ressourcen. Weitere Informationen zur Verwendung von finden Sie in der AWS CLI Befehlsreferenz.AWS CLI Weitere Informationen zur Verwendung von API-Aktionen mit Amazon CloudWatch Internet Monitor finden Sie im Amazon CloudWatch Internet Monitor API-Referenzhandbuch.

Themen

- Einen Monitor erstellen
- Überwachungsdetails anzeigen
- Auflisten von Zustandsereignissen
- Anzeigen bestimmter Zustandsereignisse
- Anzeigen der Monitorliste
- Monitor bearbeiten
- Monitor löschen

Einen Monitor erstellen

Wenn Sie in Internet Monitor einen Monitor erstellen, geben Sie einen Namen an und ordnen dem Monitor Ressourcen zu, um anzuzeigen, wo sich der Internetverkehr Ihrer Anwendung befindet. Sie geben einen Prozentsatz für den Datenverkehr an, der festlegt, wie viel von Ihrem Anwendungsverkehr überwacht wird. Das bestimmt auch die Anzahl der Stadtnetze, also der Kundenstandorte und ASNs, in der Regel Internetdienstanbieter oder ISPs, die überwacht werden. Sie können auch ein Limit für die maximale Anzahl der zu überwachenden Stadtnetze für Ihre Anwendungsressourcen festlegen, um Ihre Kosten zu kontrollieren. Weitere Informationen finden Sie unter Auswahl einer Höchstgrenze für Städtenetze.

Schließlich können Sie wählen, ob Sie alle Internetmessungen für Ihre Anwendung in Amazon S3 veröffentlichen möchten. Internet-Messungen für die 500 größten Stadtnetzwerke (nach Verkehrsaufkommen) werden von Internet Monitor automatisch in CloudWatch Logs veröffentlicht. Sie können sich jedoch dafür entscheiden, alle Messungen auch in S3 zu veröffentlichen.

Um einen Monitor mit dem zu erstellen AWS CLI, verwenden Sie den `create-monitor` Befehl. Mit dem folgenden Befehl erstellen Sie einen Monitor, der 100 % des Datenverkehrs überwacht, aber ein maximales Stadtnetz-Limit von 10 000 festlegt, eine VPC-Ressource hinzufügt und die Veröffentlichung von Internetmessungen in Amazon S3 vorsieht.

> **Note**
>
> Internet Monitor veröffentlicht in CloudWatch Logs alle fünf Minuten Internet-Messungen für die 500 wichtigsten Stadtnetzwerke (Kundenstandorte und ASNs, in der Regel Internetdienstanbieter oder ISPs), die Datenverkehr an die einzelnen Monitore senden. Optional können Sie Internetmessungen für alle überwachten Stadtnetze (bis zum Limit von 500 000 Stadtnetzen) in einem Amazon-S3-Bucket veröffentlichen. Weitere Informationen finden Sie unter Internet-Messungen in Amazon S3 in Amazon CloudWatch Internet Monitor veröffentlichen.

```
aws internetmonitor --create-monitor monitor-name "TestMonitor" \
    --traffic-percentage-to-monitor 100 \
    --max-city-networks-to-monitor 10000 \
    --resources "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-11223344556677889" \
    --internet-measurements-log-delivery
 S3Config="{BucketName=MyS3Bucket,LogDeliveryStatus=ENABLED}"
```

```
{
"Arn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
 "Status": "ACTIVE"
}
```

> **Note**
>
> Sie können den Namen eines Monitors nicht ändern.

Überwachungsdetails anzeigen

Um Informationen über einen Monitor mit dem anzuzeigen AWS CLI, verwenden Sie den Befehl. `get-monitor`

```
aws internetmonitor get-monitor --monitor-name "TestMonitor"
```

```
{
    "ClientLocationType": "city",
    "CreatedAt": "2022-09-22T19:27:47Z",
    "ModifiedAt": "2022-09-22T19:28:30Z",
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "MonitorName": "TestMonitor",
    "ProcessingStatus": "OK",
    "ProcessingStatusInfo": "The monitor is actively processing data",
    "Resources": [
        "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-11223344556677889"
    ],
    "MaxCityNetworksToMonitor": 10000,
    "Status": "ACTIVE"
}
```

Auflisten von Zustandsereignissen

Wenn die Leistung des Internetverkehrs Ihrer Anwendung nachlässt, erstellt Internet Monitor Zustandsereignisse in Ihrem Monitor. Um eine Liste der aktuellen Gesundheitsereignisse mit dem anzuzeigen AWS CLI, verwenden Sie den `list-health-events` Befehl

```
aws internetmonitor list-health-events --monitor-name "TestMonitor"
```

```
{
    "HealthEvents": [
        {
            "EventId": "2022-06-20T01-05-05Z/latency",
            "Status": "RESOLVED",
            "EndedAt": "2022-06-20T01:15:14Z",
            "ServiceLocations": [
                {
                    "Name": "us-east-1"
                }
            ],
            "PercentOfTotalTrafficImpacted": 1.21,
            "ClientLocations": [
                {
                    "City": "Lockport",
                    "PercentOfClientLocationImpacted": 60.370000000000005,
                    "PercentOfTotalTraffic": 2.01,
                    "Country": "United States",
                    "Longitude": -78.6913,
                    "AutonomousSystemNumber": 26101,
                    "Latitude": 43.1721,
                    "Subdivision": "New York",
                    "NetworkName": "YAHOO-BF1"
                }
            ],
            "StartedAt": "2022-06-20T01:05:05Z",
            "ImpactType": "PERFORMANCE",
            "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/
TestMonitor/health-event/2022-06-20T01-05-05Z/latency"
        },
        {
            "EventId": "2022-06-20T01-17-56Z/latency",
            "Status": "RESOLVED",
            "EndedAt": "2022-06-20T01:30:23Z",
            "ServiceLocations": [
                {
                    "Name": "us-east-1"
                }
            ],
            "PercentOfTotalTrafficImpacted": 1.29,
            "ClientLocations": [
                {
                    "City": "Toronto",
```

```
                    "PercentOfClientLocationImpacted": 75.32,
                    "PercentOfTotalTraffic": 1.05,
                    "Country": "Canada",
                    "Longitude": -79.3623,
                    "AutonomousSystemNumber": 14061,
                    "Latitude": 43.6547,
                    "Subdivision": "Ontario",
                    "CausedBy": {
                        "Status": "ACTIVE",
                        "Networks": [
                            {
                                "AutonomousSystemNumber": 16509,
                                "NetworkName": "Amazon.com"
                            }
                        ],
                        "NetworkEventType": "AWS"
                    },
                    "NetworkName": "DIGITALOCEAN-ASN"
                },
                {
                    "City": "Lockport",
                    "PercentOfClientLocationImpacted": 22.91,
                    "PercentOfTotalTraffic": 2.01,
                    "Country": "United States",
                    "Longitude": -78.6913,
                    "AutonomousSystemNumber": 26101,
                    "Latitude": 43.1721,
                    "Subdivision": "New York",
                    "NetworkName": "YAHOO-BF1"
                },
                {
                    "City": "Hangzhou",
                    "PercentOfClientLocationImpacted": 2.88,
                    "PercentOfTotalTraffic": 0.7799999999999999,
                    "Country": "China",
                    "Longitude": 120.1612,
                    "AutonomousSystemNumber": 37963,
                    "Latitude": 30.2994,
                    "Subdivision": "Zhejiang",
                    "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
                }
            ],
            "StartedAt": "2022-06-20T01:17:56Z",
            "ImpactType": "PERFORMANCE",
```

```
            "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/
TestMonitor/health-event/2022-06-20T01-17-56Z/latency"
        },
        {
            "EventId": "2022-06-20T01-34-20Z/latency",
            "Status": "RESOLVED",
            "EndedAt": "2022-06-20T01:35:04Z",
            "ServiceLocations": [
                {
                    "Name": "us-east-1"
                }
            ],
            "PercentOfTotalTrafficImpacted": 1.15,
            "ClientLocations": [
                {
                    "City": "Lockport",
                    "PercentOfClientLocationImpacted": 39.45,
                    "PercentOfTotalTraffic": 2.01,
                    "Country": "United States",
                    "Longitude": -78.6913,
                    "AutonomousSystemNumber": 26101,
                    "Latitude": 43.1721,
                    "Subdivision": "New York",
                    "NetworkName": "YAHOO-BF1"
                },
                {
                    "City": "Toronto",
                    "PercentOfClientLocationImpacted": 29.770000000000003,
                    "PercentOfTotalTraffic": 1.05,
                    "Country": "Canada",
                    "Longitude": -79.3623,
                    "AutonomousSystemNumber": 14061,
                    "Latitude": 43.6547,
                    "Subdivision": "Ontario",
                    "CausedBy": {
                        "Status": "ACTIVE",
                        "Networks": [
                            {
                                "AutonomousSystemNumber": 16509,
                                "NetworkName": "Amazon.com"
                            }
                        ],
                        "NetworkEventType": "AWS"
                    },
```

```
                    "NetworkName": "DIGITALOCEAN-ASN"
                },
                {
                    "City": "Hangzhou",
                    "PercentOfClientLocationImpacted": 2.88,
                    "PercentOfTotalTraffic": 0.7799999999999999,
                    "Country": "China",
                    "Longitude": 120.1612,
                    "AutonomousSystemNumber": 37963,
                    "Latitude": 30.2994,
                    "Subdivision": "Zhejiang",
                    "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
                }
            ],
            "StartedAt": "2022-06-20T01:34:20Z",
            "ImpactType": "PERFORMANCE",
            "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/
TestMonitor/health-event/2022-06-20T01-34-20Z/latency"
        }
    ]
}
```

Anzeigen bestimmter Zustandsereignisse

Um detailliertere Informationen zu einem bestimmten Zustandsereignis mit der CLI zu erhalten, führen Sie den `get-health-event`-Befehl mit Ihrem Monitornamen und einer Zustandsereignis-ID aus.

```
aws internetmonitor get-monitor --monitor-name "TestMonitor" --event-id "health-event/
TestMonitor/2021-06-03T01:02:03Z/latency"
```

```
{
    "EventId": "2022-06-20T01-34-20Z/latency",
    "Status": "RESOLVED",
    "EndedAt": "2022-06-20T01:35:04Z",
    "ServiceLocations": [
        {
            "Name": "us-east-1"
        }
    ],
    "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor/
health-event/2022-06-20T01-34-20Z/latency",
```

```
    "LastUpdatedAt": "2022-06-20T01:35:04Z",
    "ClientLocations": [
        {
            "City": "Lockport",
            "PercentOfClientLocationImpacted": 39.45,
            "PercentOfTotalTraffic": 2.01,
            "Country": "United States",
            "Longitude": -78.6913,
            "AutonomousSystemNumber": 26101,
            "Latitude": 43.1721,
            "Subdivision": "New York",
            "NetworkName": "YAHOO-BF1"
        },
        {
            "City": "Toronto",
            "PercentOfClientLocationImpacted": 29.770000000000003,
            "PercentOfTotalTraffic": 1.05,
            "Country": "Canada",
            "Longitude": -79.3623,
            "AutonomousSystemNumber": 14061,
            "Latitude": 43.6547,
            "Subdivision": "Ontario",
            "CausedBy": {
                "Status": "ACTIVE",
                "Networks": [
                    {
                        "AutonomousSystemNumber": 16509,
                        "NetworkName": "Amazon.com"
                    }
                ],
                "NetworkEventType": "AWS"
            },
            "NetworkName": "DIGITALOCEAN-ASN"
        },
        {
            "City": "Shenzhen",
            "PercentOfClientLocationImpacted": 4.07,
            "PercentOfTotalTraffic": 0.61,
            "Country": "China",
            "Longitude": 114.0683,
            "AutonomousSystemNumber": 37963,
            "Latitude": 22.5455,
            "Subdivision": "Guangdong",
            "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
```

```
        },
        {
            "City": "Hangzhou",
            "PercentOfClientLocationImpacted": 2.88,
            "PercentOfTotalTraffic": 0.7799999999999999,
            "Country": "China",
            "Longitude": 120.1612,
            "AutonomousSystemNumber": 37963,
            "Latitude": 30.2994,
            "Subdivision": "Zhejiang",
            "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
        }
    ],
    "StartedAt": "2022-06-20T01:34:20Z",
    "ImpactType": "PERFORMANCE",
    "PercentOfTotalTrafficImpacted": 1.15
}
```

Anzeigen der Monitorliste

Führen Sie den `list-monitors`-Befehl aus, um eine Liste aller Monitore in Ihrem Konto mit der CLI anzuzeigen.

```
aws internetmonitor list-monitors
```

```
{
    "Monitors": [
        {
            "MonitorName": "TestMonitor",
            "ProcessingStatus": "OK",
            "Status": "ACTIVE"
        }
    ],
    "NextToken": " zase12"
}
```

Monitor bearbeiten

Um Informationen über Ihren Monitor mithilfe der CLI zu aktualisieren, verwenden Sie den `update-monitor`-Befehl und geben Sie den Namen des zu aktualisierenden Monitors an. Sie können den Prozentsatz des zu überwachenden Datenverkehrs aktualisieren, die maximale Anzahl der zu

überwachenden Stadtnetze begrenzen, die Ressourcen, die Internet Monitor zur Überwachung des Datenverkehrs verwendet, hinzufügen oder entfernen und den Überwachungsstatus von `ACTIVE` auf `INACTIVE`, oder umgekehrt, ändern. Beachten Sie, dass Sie den Namen des Monitors nicht mehr ändern können.

Die Antwort auf einen `update-monitor`-Aufruf gibt nur den `MonitorArn` und den `Status` zurück.

Das folgende Beispiel zeigt, wie Sie den Befehl `update-monitor` verwenden, um die maximale Anzahl der zu überwachenden Stadtnetze auf `50000` zu ändern:

```
aws internetmonitor update-monitor --monitor-name "TestMonitor" --max-city-networks-to-
monitor 50000
```

```
{
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "Status": " ACTIVE "
}
```

Das folgende Beispiel zeigt, wie Sie Ressource hinzufügen und entfernen:

```
aws internetmonitor update-monitor --monitor-name "TestMonitor" \
    --resources-to-add "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-11223344556677889" \
    --resources-to-remove "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-2222444455556666"
```

```
{
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "Status": "ACTIVE"
}
```

Das folgende Beispiel zeigt, wie der Befehl `update-monitor` verwendet wird, um den Monitorstatus auf `INACTIVE` zu ändern:

```
aws internetmonitor update-monitor --monitor-name "TestMonitor" --status "INACTIVE"
```

```
{
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "Status": "INACTIVE"
}
```

Monitor löschen

Sie können einen Monitor mit der CLI löschen, indem Sie den `delete-monitor`-Befehl verwenden. Zunächst müssen Sie den Monitor auf inaktiv einstellen. Verwenden Sie den `update-monitor`-Befehl, um den Status auf `INACTIVE` zu ändern. Vergewissern Sie sich, dass der Monitor inaktiv ist, indem Sie den `get-monitor`-Befehl verwenden und den Status überprüfen.

Wenn der Monitorstatus `INACTIVE` ist, können Sie die CLI verwenden, um den `delete-monitor`-Befehl zum Löschen des Monitors auszuführen. Die Antwort auf einen erfolgreichen `delete-monitor`-Aufruf ist leer.

```
aws internetmonitor delete-monitor --monitor-name "TestMonitor"
```

```
{}
```

Überwachen und optimieren mit dem Internet-Monitor-Dashboard

Die Informationen in diesem Abschnitt beschreiben, wie Sie Informationen im Amazon CloudWatch Internet Monitor-Dashboard filtern und anzeigen können, um den Internetverkehr und die Einrichtung Ihrer AWS Anwendung zu visualisieren und Einblicke zu erhalten.

Nachdem Sie einen Monitor zur Überwachung der Internetleistung und -verfügbarkeit Ihrer Anwendung erstellt haben, veröffentlicht Amazon CloudWatch Internet Monitor CloudWatch Protokolle mit Internet-Messungen für Paare zwischen Client-Standort und Netzwerk (Stadt-Netzwerk) und veröffentlicht aggregierte CloudWatch Metriken für den Datenverkehr zu Ihrer Anwendung sowie zu jedem AWS-Region Edge-Standort. Sie können diese Informationen von Internet Monitor auf verschiedene Arten filtern, untersuchen und handlungsorientierte Vorschläge daraus erhalten.

Wählen Sie zunächst auf der CloudWatch Konsole unter Netzwerküberwachung die Option Internet Monitor aus.

In diesem Abschnitt wird hauptsächlich beschrieben, wie Sie Internet Monitor-Metriken mithilfe von filtern und anzeigen AWS Management Console. Alternativ können Sie Internet Monitor-API-Operationen mit dem AWS CLI oder einem SDK verwenden, um direkt mit Internet Monitor-Ereignissen zu arbeiten, die in CloudWatch Protokolldateien gespeichert sind. Weitere Informationen finden Sie unter Verwenden Ihrer Monitor- und Messinformationen. Weitere Informationen zur

Verwendung von API-Vorgängen finden Sie unter Beispiele für die Verwendung der CLI mit Amazon CloudWatch Internet Monitor und in der Amazon CloudWatch Internet Monitor-API-Referenz.

Das Internet-Monitor-Dashboard enthält drei Registerkarten:

- Auf der Registerkarte Overview (Übersicht) finden Sie aktuelle und historische Leistungs- und Verfügbarkeitsinformationen zu Ihrer Anwendung sowie Zustandsereignisse, die sich auf Ihre Kundenstandorte auswirken.
- Auf der nächsten Registerkarte Explorerverlauf können Sie nach Standort, ASN, Datum usw. filtern und mithilfe der Grafiken die Metriken für Ihren Internetverkehr im Laufe der Zeit visualisieren.
- Auf der Registerkarte Datenverkehrseinblicke können Sie sich nicht nur Informationen über den am häufigsten überwachten Datenverkehr anzeigen lassen, die auf verschiedene anpassbare Arten zusammengefasst sind, sondern auch Vorschläge für optimierte Setups erhalten, um die Leistung für verschiedene Standort- und ASN-Paare zu verbessern. Internet Monitor prognostiziert die Leistungsverbesserung Ihrer Anwendung auf der Grundlage Ihrer Datenverkehrsmuster und der Leistung in der Vergangenheit, wenn Sie die Art und Weise ändern, wie Sie Ihren Datenverkehr weiterleiten oder welche AWS Ressourcen Sie verwenden. Sie können auch in einem Diagramm vergleichen, wie viele Stadtnetze in Ihrer Überwachungsabdeckung enthalten sind, basierend auf dem Prozentsatz des Anwendungsverkehrs, den Sie für Ihren Monitor ausgewählt haben.

Da Internet Monitor Protokolldateien mit den Messungen Ihres Datenverkehrs generiert und veröffentlicht, können Sie außerdem andere CloudWatch Tools in der Konsole verwenden, um die von Internet Monitor veröffentlichten Daten weiter zu visualisieren, darunter CloudWatch Contributor Insights, CloudWatch Metrics und CloudWatch Logs Insights. Weitere Informationen finden Sie unter Erkunden Sie Ihre Daten mit CloudWatch Tools und der Internet Monitor-Abfrageschnittstelle.

In den folgenden Abschnitten erfahren Sie, wie Sie Internet Monitor verwenden, um Ihre Leistungs- und Verfügbarkeitsmessungen zu untersuchen.

Themen

- Verfolgen von Leistung und Verfügbarkeit in Echtzeit in Amazon CloudWatch Internet Monitor (Registerkarte „Übersicht“)
- Filtern und Anzeigen von historischen Daten in Amazon CloudWatch Internet Monitor (Registerkarte Historischer Explorer)
- Erhalten von Erkenntnissen zur Verbesserung der Anwendungsleistung in Amazon CloudWatch Internet Monitor (Registerkarte Traffic Insights)

Verfolgen von Leistung und Verfügbarkeit in Echtzeit in Amazon CloudWatch Internet Monitor (Registerkarte „Übersicht“)

Verwenden Sie die Registerkarte Übersicht in der CloudWatch Konsole unter Internet Monitor, um sich einen Überblick über die Leistung und Verfügbarkeit des Datenverkehrs zu verschaffen, den Ihr Monitor verfolgt. Die Registerkarte zeigt auch eine Übersichtskarte des Internetverkehrs mit Verkehrsclustern, die Ihnen helfen können, den globalen Verkehr Ihrer Anwendung sowie den Ort und die Auswirkungen von Zustandsereignissen zu visualisieren.

Zustandsbewertungen

Das Diagramm mit den Gesundheitswerten zeigt Ihnen Leistungs- und Verfügbarkeitsinformationen für Ihren weltweiten Traffic. AWS enthält umfangreiche historische Daten zur Internetleistung und -verfügbarkeit für den Netzwerkverkehr zwischen geografischen Standorten für verschiedene ASNs und AWS Dienste. Internet Monitor verwendet die Verbindungsdaten, die im AWS Rahmen seiner globalen Netzwerkauslastung erfasst wurden, um eine Ausgangsbasis für Leistung und Verfügbarkeit des Internetverkehrs zu berechnen. Dies sind dieselben Daten, die wir verwenden, AWS um unsere eigene Internetverfügbarkeit und -verfügbarkeit zu überwachen.

Mit diesen Messungen als Ausgangswert kann Internet Monitor erkennen, wann Leistung und Verfügbarkeit Ihrer Anwendung im Vergleich zum Ausgangswert gesunken sind. Damit Sie diese Rückgänge leichter erkennen können, teilen wir Ihnen diese Informationen in Form einer Leistungs- und einer Verfügbarkeitsbewertung mit. Weitere Informationen finden Sie unter Erkunden Sie Ihre Daten mit CloudWatch Tools und der Internet Monitor-Abfrageschnittstelle.

Das Diagramm mit den Zustandsbewertungen enthält Zustandsereignisse, die in einem von Ihnen ausgewählten Zeitraum aufgetreten sind. Bei einem Zustandsereignis sehen Sie in der Grafik, wie die Leistungs- oder Verfügbarkeitslinie abfällt. Wenn Sie das Ereignis auswählen, werden weitere Details und Bänder in der Grafik angezeigt. Datums- und Uhrzeitinformationen geben an, wie lange das Ereignis gedauert hat.

Sie können sich diese Metriken auch ansehen, indem Sie direkt auf die Protokolldateien für jeden Datenpunkt zugreifen. Wählen Sie im Menü „Aktionen“ die Option „ CloudWatch Protokolle anzeigen“.

Übersicht über den Internetverkehr

Die Karte Internetverkehrs-Übersicht zeigt Ihnen den Internetverkehr und die Zustandsereignisse, die sich auf die Standorte und ASNs beziehen, von denen aus Ihre Benutzer auf Ihre Anwendung

zugreifen. Die Länder, die auf der Karte grau dargestellt sind, sind diejenigen, die Verkehr für Ihre Anwendung enthalten.

Jeder Kreis auf der Karte weist auf ein Zustandsereignis in einem Gebiet für einen von Ihnen ausgewählten Zeitraum hin. Internet Monitor erstellt Integritätsereignisse, wenn er bei einem bestimmten Schwellenwert ein Problem mit der Konnektivität zwischen einer Ihrer Ressourcen, in der gehostet wird, AWS und einem Stadtnetzwerk, in dem ein Benutzer auf Ihre Anwendung zugreift, feststellt. Wenn Sie auf der Karte einen Kreis auswählen, werden weitere Details zum Zustandsereignis für diesen Ort angezeigt. Darüber hinaus finden Sie für Cluster mit Zustandsereignissen detaillierte Informationen in der Tabelle Health events (Zustandsereignisse) unterhalb der Karte.

Beachten Sie, dass Internet Monitor Zustandsereignisse in einem Monitor erstellt, wenn er feststellt, dass ein Ereignis erhebliche globale Auswirkungen auf Ihre Anwendung hat. Wenn in dem von Ihnen gewählten Zeitraum keine Zustandsereignisse auftreten, die den Schwellenwert für die Auswirkung auf den Verkehr an den Kundenstandorten überschreiten, ist die Karte leer. Weitere Informationen finden Sie unter Wann erstellt Internet Monitor Zustandsereignisse und löst sie auf.

Schwellenwerte für Zustandsereignisse ändern

Sie können verschiedene Optionen dafür konfigurieren, wie und wann Internet Monitor Zustandsereignisse für Ihre Anwendung erstellt. Wählen Sie Schwellenwerte aktualisieren, um Änderungen vorzunehmen.

Sie können den allgemeinen Schwellenwert ändern, der Internet Monitor dazu veranlasst, ein Zustandsereignis zu erzeugen. Der Standardschwellenwert für Zustandsereignisse liegt sowohl für Leistungs- als auch für Verfügbarkeitsbewertungen bei 95 %. Das bedeutet, dass Internet Monitor ein Zustandsereignis erzeugt, wenn die Gesamtleistung oder die Verfügbarkeit Ihrer Anwendung auf 95 % oder darunter fällt. Für den Gesamtschwellenwert kann das Zustandsereignis durch ein einzelnes großes Problem oder durch die Kombination mehrerer kleinerer Probleme ausgelöst werden.

Sie können auch den lokalen, also Stadtnetz-Schwellenwert in Kombination mit einem Prozentsatz der Gesamtauswirkung ändern, der ein Zustandsereignis auslöst. Wenn Sie einen Schwellenwert festlegen, der ein Zustandsereignis auslöst, wenn ein Wert unter den Schwellenwert für ein oder mehrere Stadtnetze (Standorte und ASNs, in der Regel ISPs) fällt, erhalten Sie Einblicke, wenn es beispielsweise an Standorten mit geringerem Datenverkehr Probleme gibt.

Eine zusätzliche lokale Schwellenwertoption funktioniert zusammen mit dem lokalen Schwellenwert für Verfügbarkeits- oder Leistungswerte. Der zweite Faktor ist der Prozentsatz Ihres gesamten Datenverkehrs, der betroffen sein muss, bevor Internet Monitor ein Zustandsereignis auf der Grundlage des lokalen Schwellenwerts erzeugt.

Durch die Konfiguration der Schwellenwertoptionen für den Gesamtverkehr und den lokalen Verkehr können Sie die Häufigkeit der Erstellung von Zustandsereignissen genau auf die Nutzung Ihrer Anwendung und Ihre Bedürfnisse abstimmen. Beachten Sie, dass, wenn Sie den lokalen Schwellenwert niedriger einstellen, in der Regel mehr Zustandsereignisse erzeugt werden, je nach Ihrer Anwendung und den anderen von Ihnen eingestellten Konfigurationswerten für den Schwellenwert.

Zusammenfassend können Sie die Schwellenwerte für Zustandsereignisse – für Leistungswerte, Verfügbarkeitswerte oder beides – auf folgende Weise konfigurieren:

- Auswahl verschiedener globaler Schwellenwerte für die Auslösung eines Zustandsereignisses.
- Auswahl verschiedener lokaler Schwellenwerte für die Auslösung eines Zustandsereignisses. Mit dieser Option können Sie auch den Prozentsatz der Auswirkungen auf Ihre Gesamtanwendung ändern, der überschritten werden muss, bevor Internet Monitor ein Ereignis erzeugt.
- Wählen Sie, ob Sie das Auslösen eines Zustandsereignisses auf der Grundlage lokaler Schwellenwerte deaktivieren oder lokale Schwellenwertoptionen aktivieren möchten.

Sie können auch Optionen für Leistungs- oder Verfügbarkeitswerte oder für beides konfigurieren. Sie können eine Kombination der Optionen oder nur eine davon konfigurieren.

Gehen Sie wie folgt vor, um Schwellenwerte und andere Konfigurationsoptionen für Leistungswerte, Verfügbarkeitswerte oder beides zu aktualisieren:

So ändern Sie die Konfigurationsoptionen für Schwellenwerte

1. Navigieren Sie im AWS Management Console zu Internet Monitor CloudWatch, und wählen Sie dann im linken Navigationsbereich aus.
2. Wählen Sie auf der Registerkarte Übersicht im Abschnitt Zeitleiste der Zustandsereignisse die Option Schwellenwerte aktualisieren.
3. Wählen Sie auf der sich öffnenden Dialogseite die neuen Werte und Optionen aus, die Sie für Schwellenwerte und andere Optionen wünschen, die Internet Monitor zur Erstellung eines Zustandsereignisses veranlassen. Sie können einen der folgenden Schritte ausführen:

- Wählen Sie einen neuen Wert für den Schwellenwert für die Verfügbarkeitsbewertung, den Schwellenwert für die Leistungsbewertung oder beide.

  Die Diagramme in den Abschnitten für die jeweiligen Einstellungen zeigen die aktuelle Schwellenwerteinstellung und die tatsächlichen jüngsten Zustandsereignisse für die Verfügbarkeit oder Leistung Ihrer Anwendung an. Wenn Sie sich die typischen Werte ansehen, können Sie sich ein Bild von den Werten machen, auf die Sie einen Schwellenwert ändern möchten.

  Tipp: Um ein größeres Diagramm anzuzeigen und den Zeitrahmen zu ändern, wählen Sie den Expander in der oberen rechten Ecke des Diagramms.
- Wählen Sie, ob Sie einen lokalen Schwellenwert für die Verfügbarkeit oder die Leistung oder beides aktivieren oder deaktivieren möchten. Wenn eine Option aktiviert ist, können Sie den Schwellenwert und die Auswirkungsstufe festlegen, wann Internet Monitor ein Zustandsereignis erstellen soll.

4. Nachdem Sie die Schwellenwertoptionen konfiguriert haben, speichern Sie Ihre Aktualisierungen, indem Sie Schwellenwerte für Zustandsereignisse aktualisieren wählen.

Weitere Informationen über die Funktionsweise von Zustandsereignissen finden Sie unter Wann erstellt Internet Monitor Zustandsereignisse und löst sie auf.

Tabelle Zustandsereignisse

In der Tabelle Zustandsereignisse sind Kundenstandorte aufgeführt, die von Zustandsereignissen betroffen waren, sowie Informationen zu den Ereignissen. Die folgenden Spalten sind in der Tabelle enthalten.

	Beschreibung
Kundenstandort	Dies ist der Standort der Endbenutzer, die von dem Ereignis betroffen waren und bei denen eine erhöhte Latenz oder eine verringerte Verfügbarkeit zu verzeichnen war. Weitere Informationen zur Genauigkeit des Standorts von Kunden in Internet Monitor finden Sie unter Geolokalisierungsinformationen und Genauigkeit in Internet Monitor.

	Beschreibung
Auswirkungen auf den Datenverkehr	Die Stärke der Auswirkung, die Ereignis in Form einer erhöhten Latenz oder einer verringerten Verfügbarkeit verursacht hat. Bei der Latenz handelt es sich um den Prozentsatz, um den sich die Latenz während des Ereignisses im Vergleich zur typischen Leistung für den Datenverkehr von diesem Client-Standort zu diesem AWS Standort über dieses Client-Netzwerk erhöht hat.
Client-Netzwerk	Das Netzwerk, über das der Datenverk ehr übertragen wurde. In der Regel ist dies der Internetdienstanbieter (ISP) oder die autonome Systemnnummer (ASN) für den Netzwerkverkehr.
AWS Standort	Der AWS Standort für den Netzwerkverkehr. Dabei kann es sich um einen Standort AWS-Region oder einen Internet-Edge-Standort handeln.

	Beschreibung
Art der Auswirkung	Die Art der Auswirkung auf das Zustandsereignis. Zustandsereignisse werden in der Regel durch Latenzerhöhungen (Leistungsprobleme) oder Erreichbarkeitsprobleme (Verfügbarkeitsprobleme) verursacht. Sie sollten auch auf den Auswirkungstyp klicken können, um die Ursache der Beeinträchtigung zu sehen. Wenn möglich, analysiert Internet Monitor den Ursprung eines Integritätsereignisses, um festzustellen, ob es durch AWS oder durch einen ASN (Internet Service Provider) verursacht wurde. Beachten Sie, dass diese Analyse fortgesetzt wird, nachdem das Ereignis behoben ist. Internet Monitor kann Ereignisse mit neuen Informationen bis zu eine Stunde lang aktualisieren.

Wenn Sie einen der Kundenstandorte in der Tabelle der Zustandsereignisse auswählen, sehen Sie weitere Details zu dem Zustandsereignis an diesem Standort. Sie können beispielsweise sehen, wann das Ereignis begonnen hat, wann es geendet hat und wie sich das auf den lokalen Datenverkehr ausgewirkt hat.

Visualisierung von Netzwerkpfaden

Eine abgeschlossene Beeinträchtigungsanalyse hat einen vollständigen Netzwerkpfad unter Netzwerkpfad-Visualisierung. Der vollständige Pfad zeigt Ihnen jeden Knoten entlang des Netzwerkpfads für Ihre Anwendung für das Integritätsereignis, zwischen dem AWS Standort und dem Client, für ein Client-Standort-Paar.

Wenn Internet Monitor die Ursache für eine Beeinträchtigung feststellt, wird diese mit einem gestrichelten roten Kreis markiert. Beeinträchtigungen können durch ASNs, in der Regel Internetdienstanbieter (ISP), verursacht werden, oder die Ursache kann AWS sein. Wenn es mehrere Ursachen für eine Beeinträchtigung gab, werden mehrere Knoten eingekreist.

Filtern und Anzeigen von historischen Daten in Amazon CloudWatch Internet Monitor (Registerkarte Historischer Explorer)

Verwenden Sie die Registerkarte Historischer Explorer in der CloudWatch Konsole unter Internet Monitor, um Daten für Ihre Anwendung, die sich in den CloudWatch Protokollen befinden, zu filtern und anzuzeigen. Internet Monitor veröffentlicht Messungen in anwendungsspezifischen CloudWatch Protokollen zur Verfügbarkeit, Leistung, übertragenen überwachten Byte (oder Anzahl der Client-Verbindungen, nur für WorkSpaces Verzeichnisse) und Round-Trip-Zeit für Ihre überwachten Stadtnetzwerke in AWS-Regionen.

> **Note**
>
> Internet Monitor veröffentlicht alle fünf Minuten Internet-Messwerte in CloudWatch Logs für die 500 (nach Verkehrsaufkommen) größten Stadtnetzwerke (d. h. Kundenstandorte und ASNs, in der Regel Internetdienstanbieter oder ISPs), die Datenverkehr an die einzelnen Monitore senden. Optional können Sie Internetmessungen für alle überwachten Stadtnetze (bis zum Limit von 500 000 Stadtnetzen) in einem Amazon-S3-Bucket veröffentlichen. Weitere Informationen finden Sie unter Internet-Messungen in Amazon S3 in Amazon CloudWatch Internet Monitor veröffentlichen.

Um mit der Untersuchung Ihrer Anwendungsdaten zu beginnen, wählen Sie eine Zeitspanne aus. Wählen Sie dann einen bestimmten geografischen Ort, beispielsweise eine Stadt, und (optional) weitere Filter. Internet Monitor wendet die Filter auf die Daten in den Internetmessungsprotokollen an, die er für die Stadtnetze für Ihren Anwendungsverkehr veröffentlicht hat. Anschließend können Sie Diagramme mit den Daten anzeigen, die den Leistungswert, die Verfügbarkeitsbewertung, die übertragenen überwachten Byte (für VPCs, Network Load Balancer und CloudFront Verteilungen) oder die Anzahl der Client-Verbindungen (für WorkSpaces Verzeichnisse) und die Round-Trip-Zeit (RTT) für Ihre Anwendung im Zeitverlauf zeigen.

Die Tabelle All events (Alle Ereignisse) unter den Grafiken zeigt Ihnen die Zustandsereignisse, die Ihr Filter für Ihren Anwendungsdatenverkehr zurückgibt, sowie Informationen zu jedem Ereignis. Das umfasst die folgenden Spalten.

	Beschreibung
Beginn des Ereignisses	Der Zeitpunkt, zu dem das Zustandsereignis begann.
Status	Ob das Ereignis noch aktiv ist oder gelöst ist.
Kundenstandort	Der Standort der Endbenutzer, die von dem Ereignis betroffen waren und bei denen eine erhöhte Latenz oder eine verringerte Leistung zu verzeichnen war. Weitere Informationen zur Genauigkeit des Standorts von Kunden in Internet Monitor finden Sie unter Geolokalisierungsinformationen und Genauigkeit in Internet Monitor.
Auswirkungen auf den Datenverkehr	Die gewichteten Auswirkungen des Ereignisses auf den Ort des Zustandsereignisses. Dies ist beispielsweise die Auswirkung auf die Latenz im Vergleich zur typischen Leistung für den Datenverkehr von einem Kundenstandort zum Standort über die AWS Client-ASN, in der Regel ein Internetdienstanbieter (ISP). Ähnlich sehen Sie bei einem Ereignis, das sich auf die Verfügbarkeit auswirkt, die Auswirkungen auf die Verfügbarkeit im Vergleich zur typischen Verfügbarkeit für den Client-Standort für den AWS Standort über die Client-ASN.
Dauer des Ereignisses	Gibt an, wie lange das Ereignis andauerte. Internet Monitor beendet Zustandsereignisse, wenn sie nicht mehr als (insgesamt) 5 % der Kundenstandorte Ihrer Anwendung betreffen.
Client-ISP	Der ASN, in der Regel der Internetdienstanbieter (ISP), der der Träger des Netzwerkverkehrs war.

	Beschreibung
Standort des Services	Der Dienststandort, von dem der Netzwerkv erkehr ausgegangen ist. Dabei kann es sich um einen Standort AWS-Region oder einen Internet-Edge-Standort handeln.

Sie können sich die Messungen Ihrer Anwendung ansehen, indem Sie direkt auf die Protokolle für jeden Datenpunkt zugreifen. Wählen Sie im Menü Aktionen die Option CloudWatch Protokolle anzeigen aus. Beachten Sie, dass Messereignisse bei ihrer Erstellung in Ihrem Konto veröffentlicht werden, sodass Sie auch andere CloudWatch Dashboards oder Alarme auf deren Grundlage erstellen können. Weitere Informationen finden Sie unter Erhalten von Erkenntnissen zur Verbesserung der Anwendungsleistung in Amazon CloudWatch Internet Monitor (Registerkarte Traffic Insights) und Alarme mit Amazon CloudWatch Internet Monitor erstellen.

Neben der Untersuchung und Analyse von Messungen und Metriken von Internet Monitor und der Erstellung von Dashboards und Alarmen auf der Grundlage dieser Daten können Sie Internet Monitor auch dazu verwenden, zu verstehen, wie Sie die Leistung Ihrer Anwendung verbessern können. Auf der Registerkarte Traffic Insights (Datenverkehrseinblicke) finden Sie verschiedene Möglichkeiten, Optionen zu erkunden. Weitere Informationen finden Sie unter Vorschläge zur Verkehrsoptimierung auf der Registerkarte Datenverkehrseinblicke. Darüber hinaus finden Sie die spezifischen Beispiele im Kapitel Anwendungsfälle von Internet Monitor.

Erhalten von Erkenntnissen zur Verbesserung der Anwendungsleistung in Amazon CloudWatch Internet Monitor (Registerkarte Traffic Insights)

Verwenden Sie den Tab Traffic Insights in der CloudWatch Konsole unter Internet Monitor, um sich zusammenfassende Informationen über den höchsten Traffic (nach Volumen) für Ihre Anwendung anzusehen. Sie können Ihren Anwendungsdatenverkehr auf verschiedene Arten filtern und sortieren. Scrollen Sie dann nach unten und wählen Sie verschiedene Einrichtungskombinationen für Ihre Anwendung, um zu sehen, welche Alternativen Internet Monitor vorschlägt, um die schnellste Zeit bis zum ersten Byte (TTFB) zu erreichen.

Internet Monitor veröffentlicht in CloudWatch Logs alle fünf Minuten Internet-Messungen für die 500 größten Stadtnetzwerke (d. h. Kundenstandorte und ASNs, in der Regel Internetdienstanbieter oder ISPs), die Datenverkehr an die einzelnen Monitore senden. Optional können Sie Internetmessungen für alle überwachten Stadtnetze (bis zum Limit von 500 000 Stadtnetzen) in einem Amazon-S3-

Bucket veröffentlichen. Weitere Informationen finden Sie unter Internet-Messungen in Amazon S3 in Amazon CloudWatch Internet Monitor veröffentlichen.

Die wichtigsten Datenverkehrszusammenfassungen

Sie können damit beginnen, Zusammenfassungen des gesamten Datenverkehrs und der Leistung Ihrer Anwendung über einen bestimmten Zeitraum, gefiltert nach dem Standort des Kunden, zu betrachten. Sie können auch die Leistung Ihrer Anwendung für die wichtigsten (oder unwichtigsten) Kundenstandorte nach Verkehrsaufkommen betrachten, gefiltert und auf verschiedene Weise sortiert. Sie können beispielsweise nach Granularität (also nach Stadt, Unterbezirk, Land oder Ballungsraum), nach Gesamtverkehr, durchschnittlicher Zeit bis zum ersten Byte (TTFB) und anderen Faktoren sortieren.

Weitere Informationen zur Genauigkeit des Standorts von Kunden in Internet Monitor finden Sie unter Geolokalisierungsinformationen und Genauigkeit in Internet Monitor.

> **Note**
>
> Die Filter, die Sie verwenden, gelten für die gesamte Seite, so dass sie sich darauf auswirken, welche Stadtnetze in den Übersichtsgrafiken und den Informationen zum Gesamtverkehr enthalten sind und welche Stadtnetze im folgenden Abschnitt Vorschläge zur Verkehrsoptimierung enthalten sind.

Vorschläge zur Datenverkehrsoptimierung

Im Bereich Vorschläge zur Verkehrsoptimierung wird ein gefilterter Satz überwachter Stadtnetze (Standorte und ASNs, Internetdienstanbieter) für Ihren Datenverkehr zusammen mit dem gesamten Kundenverkehr für jedes einzelne Netzwerk angezeigt. Die Einträge in der Tabelle basieren auf den Filtern, die Sie für Ihren Anwendungsdatenverkehr für Datenverkehrseinblicke oben auf der Seite ausgewählt haben. Standardmäßig werden die 10 Städte mit dem höchsten Verkehrsaufkommen angezeigt. In der Regel werden in der Tabelle mehr als 10 Zeilen angezeigt, da für jedes einzelne Stadtnetz-Paar ein Eintrag vorhanden ist. Das heißt, es gibt eine Zeile für jede Kombination aus Standort (Stadt) und ASN (Netzwerkanbieter), über die Kunden auf Ihre Anwendung zugreifen, z. B. Dallas, Texas, USA und Comcast.

> **Note**
>
> Um Vorschläge zur Optimierung des Datenverkehrs für all Ihre überwachten Stadtnetzwerke zu erhalten, können Sie eine Abfrage direkt in Insights ausführen. CloudWatch Eine Beispielabfrage, die den Filter für geografische Granularität, der die Liste der Stadtnetze auf dieser Seite einschränkt, nicht enthält, finden Sie unter CloudWatch Logs Insights mit Amazon CloudWatch Internet Monitor verwenden.

Wählen Sie in diesem Abschnitt verschiedene Optionen aus: Amazon EC2 oder beide. CloudFront Auf diese Weise können Sie sehen, wie hoch die prognostizierten durchschnittlichen Werte für die durchschnittliche Zeit bis zum ersten Byte (Time to First Byte) für Clients sind, wenn Sie Ihre Anwendung mit diesen Services in verschiedenen AWS Regionen verwenden, verglichen mit dem aktuellen TTFB. Weitere Informationen zu TTFB-Berechnungen finden Sie unter AWS Berechnungen für TTFB und Latenz.

Indem Sie verschiedene Optionen auswählen und dann die Ergebnisse in der Tabelle anzeigen, können Sie mit der Planung von Setups und Implementierungen beginnen, die die Leistung für Ihre Kunden verbessern können. Beachten Sie, dass in einer Spalte möglicherweise ein Gedankenstrich (-) anstelle eines Werts angezeigt wird, wenn keine Daten zur Anzeige verfügbar sind. Ein konkretes Beispiel dafür, wie Sie die Leistung verbessern können, finden Sie unter Amazon CloudWatch Internet Monitor für ein besseres Spielerlebnis verwenden.

Experimentieren Sie zunächst für ein bestimmtes Stadtnetzwerk (Client-Standort und ASN-Paar) damit, entweder EC2 oder die CloudFront Option oder beides auszuwählen. Internet Monitor zeigt Ihnen für jedes in der Tabelle aufgelistete Stadtnetz die potenziellen Leistungsverbesserungen des TTFB, basierend auf der Wahl der Verkehrsweiterleitung (über ein bestimmtes AWS-Region) mit dieser Option im Vergleich zur aktuellen Konfiguration. (Beachten Sie, dass die Tabelle der Vollständigkeit halber auch Routen enthält, die bereits optimiert wurden.) Beispielsweise könnte Ihnen ein prognostizierter durchschnittlicher TTFB von 50 ms für die Verwendung von EC2-Routing durch `us-east-1` angezeigt werden, verglichen mit Ihrem aktuellen Setup mit einem TTFB von 100 ms, bei dem Sie EC2-Routing durch `us-west-2` verwenden. Sie könnten also das Routing durch `us-west-2`. in Betracht ziehen.

Als weiteres Beispiel könnten Sie EC2 auswählen und dann feststellen, dass es keinen messbaren Leistungsunterschied für einen Client-Standort und eine ASN macht. Beachten Sie dann jedoch, dass die TTFB etwas gesenkt wird, wenn Sie CloudFront mit derselben Region auswählen. Dies deutet darauf hin, dass das Hinzufügen einer CloudFront Distribution vor Ihrer

Anwendung zu einer Leistungsverbesserung führen kann und es sich für diesen Kundenstandort und die ASN lohnen könnte, es auszuprobieren.

Erkunden Sie Ihre Daten mit CloudWatch Tools und der Internet Monitor-Abfrageschnittstelle

Neben der Visualisierung Ihrer Leistung und Verfügbarkeit für Ihre Anwendung mit dem Amazon CloudWatch Internet Monitor-Dashboard gibt es mehrere Methoden, mit denen Sie tiefer in die Daten eintauchen können, die Internet Monitor für Sie generiert. Zu diesen Methoden gehören die Verwendung von CloudWatch Tools mit Internet Monitor-Daten, die in CloudWatch Protokolldateien gespeichert sind, und die Verwendung der Internet Monitor-Abfrageschnittstelle. Zu den Tools, die Sie verwenden können, gehören CloudWatch Logs Insights, CloudWatch Metrics, CloudWatch Contributor Insights und Amazon Athena. Je nach Bedarf können Sie einige oder alle dieser Tools sowie das Dashboard verwenden, um Internet-Monitor-Daten zu untersuchen.

Internet Monitor aggregiert CloudWatch Metriken über den Datenverkehr zu Ihrer Anwendung und zu den einzelnen AWS-Region Anwendungen und umfasst Daten wie die Gesamtbelastung des Datenverkehrs, die Verfügbarkeit und die Round-Trip-Zeit. Diese Daten werden in CloudWatch Logs veröffentlicht und können auch mit der Internet Monitor-Abfrageschnittstelle verwendet werden. Die Einzelheiten zur Geogranularität und zu anderen Aspekten der Informationen, die für einzelne Bereiche zur Verfügung stehen, sind unterschiedlich.

Amazon CloudWatch Internet Monitor veröffentlicht Daten für Ihren Monitor in Intervallen von 5 Minuten und stellt die Daten dann auf verschiedene Weise zur Verfügung. In der folgenden Tabelle werden Szenarien für den Zugriff auf Internet-Monitor-Daten aufgeführt und die Merkmale der Daten beschrieben, die für die einzelnen Szenarien gesammelt werden.

Funktion	CloudWatch Logs	Exportieren in S3	Abfragesc hnittstelle	CloudWatch Armaturenbrett
Standardmäßig aktiviert.	Ja	Nein	Ja	Ja
Anzahl der Stadtnetze, für die Daten	Top 500 (siehe Hinweis unten)	Alle	Alle	Alle

Funktion	CloudWatch Logs	Exportieren in S3	Abfragesc hnittstelle	CloudWatch Armaturenbrett
gesammelt werden				
Datenaufb ewahrung	Benutzerg esteuert	Benutzerg esteuert	30 Tage	30 Tage
Geogranul aritäten, für die Daten gesammelt werden	Alle (Stadtnet z, Metro+Net zwerk, Unterteil ung+Netzwerk, Land+Netzwerk)	Stadtnetz	Alle (Stadtnet z, Metro+Net zwerk, Unterteil ung+Netzwerk, Land+Netzwerk)	Alle (Stadtnet z, Metro+Net zwerk, Unterteil ung+Netzwerk, Land+Netzwerk)
So können Daten abgefragt und gefiltert werden	CloudWatch Logs Insights mit Amazon CloudWatch Internet Monitor verwenden	Verwendung von Amazon Athena zur Abfrage von Internetm essungen in Amazon-S3-Protokolldateien	Verwenden der Amazon CloudWatch Internet Monitor-Abfragesch nittstelle	Überwachen und optimieren mit dem Internet-Monitor-D ashboard

Anmerkung: Die Top-500-Messungen werden für Stadtnetze erfasst; die Top-250 für Metro +Netzwerke, die Top-100 für Unterteilung+Netzwerke, die Top-50 für Land+Netzwerke.

In diesem Kapitel wird beschrieben, wie Sie Ihre Daten mithilfe von CloudWatch Tools oder der Internet Monitor-Abfrageschnittstelle abfragen und untersuchen können, sowie Beispiele für jede Methode.

Inhalt

CloudWatch Logs Insights mit Amazon CloudWatch Internet Monitor verwenden

Amazon CloudWatch Internet Monitor veröffentlicht detaillierte Messungen der Verfügbarkeit und der Round-Trip-Zeit in CloudWatch Logs, und Sie können CloudWatch Logs Insights-Abfragen verwenden, um eine Teilmenge von Protokollen nach einer bestimmten Stadt oder Region (Kundenstandort), Kunden-ASN (ISP) und Quellstandort zu filtern. AWS

Weitere Informationen zur Genauigkeit des Standorts von Kunden in Internet Monitor finden Sie unter Geolokalisierungsinformationen und Genauigkeit in Internet Monitor.

Die Beispiele in diesem Abschnitt können Ihnen helfen, CloudWatch Logs Insights-Abfragen zu erstellen, um mehr über die Messungen und Metriken Ihres eigenen Anwendungsdatenverkehrs zu erfahren. Wenn Sie diese Beispiele in CloudWatch Logs Insights verwenden, ersetzen Sie *MonitorName* durch Ihren eigenen Monitornamen.

Vorschläge zur Datenverkehrsoptimierung anzeigen

Auf der Registerkarte Einblicke in den Datenverkehr in Internet Monitor können Sie Vorschläge zur Optimierung des Datenverkehrs einsehen, gefiltert nach Standort. Um dieselben Informationen zu sehen, die im Abschnitt Vorschläge zur Traffic-Optimierung auf dieser Registerkarte angezeigt werden, jedoch ohne den Standortgranularitätsfilter, können Sie die folgende CloudWatch Logs Insights-Abfrage verwenden.

1. Navigieren Sie im AWS Management Console zu CloudWatch Logs Insights.
2. Wählen Sie für Log Group (Protokollgruppe) die Option `/aws/internet-monitor/`*monitorName*`/byCity` und `/aws/internet-monitor/`*monitorName*`/byCountry` aus und geben Sie dann einen Zeitraum an.
3. Fügen Sie die folgende Abfrage hinzu und führen Sie sie aus.

```
fields @timestamp,
clientLocation.city as @city, clientLocation.subdivision as @subdivision,
 clientLocation.country as @country,
`trafficInsights.timeToFirstByte.currentExperience.serviceName` as @serviceNameField,
concat(@serviceNameField, ` (`, `serviceLocation`, `)`) as @currentExperienceField,
concat(`trafficInsights.timeToFirstByte.ec2.serviceName`, ` (`,
 `trafficInsights.timeToFirstByte.ec2.serviceLocation`, `)`) as @ec2Field,
`trafficInsights.timeToFirstByte.cloudfront.serviceName` as @cloudfrontField,
concat(`clientLocation.networkName`, ` (AS`, `clientLocation.asn`, `)`) as @networkName
| filter ispresent(`trafficInsights.timeToFirstByte.currentExperience.value`)
```

```
| stats avg(`trafficInsights.timeToFirstByte.currentExperience.value`) as @averageTTFB,
avg(`trafficInsights.timeToFirstByte.ec2.value`) as @ec2TTFB,
avg(`trafficInsights.timeToFirstByte.cloudfront.value`) as @cloudfrontTTFB,
sum(`bytesIn` + `bytesOut`) as @totalBytes,
latest(@ec2Field) as @ec2,
latest(@currentExperienceField) as @currentExperience,
latest(@cloudfrontField) as @cloudfront,
count(*) by @networkName, @city, @subdivision, @country
| display @city, @subdivision, @country, @networkName, @totalBytes, @currentExperience,
 @averageTTFB, @ec2, @ec2TTFB, @cloudfront, @cloudfrontTTFB
| sort @totalBytes desc
```

Internetverfügbarkeit und RTT anzeigen (p50, p90 und p95)

Um die Internetverfügbarkeit und die Round-Up-Zeit (p50, p90 und p95) für den Datenverkehr anzuzeigen, können Sie die folgende CloudWatch Logs Insights-Abfrage verwenden.

Region des Endbenutzers: Chicago, IL, Vereinigte Staaten

Endbenutzernetzwerk (ASN): AS7018

AWS Servicestandort: Region USA Ost (Nord-Virginia)

Um die Protokolle anzuschauen, gehen Sie wie folgt vor:

1. Navigieren Sie in der AWS Management Console zu CloudWatch Logs Insights.
2. Wählen Sie für Log Group (Protokollgruppe) die Option `/aws/internet-monitor/`*`monitorName`*`/byCity` und `/aws/internet-monitor/`*`monitorName`*`/byCountry` aus und geben Sie dann einen Zeitraum an.
3. Fügen Sie die folgende Abfrage hinzu und führen Sie sie aus.

Die Abfrage gibt alle Leistungsdaten für Benutzer zurück, die im ausgewählten Zeitraum eine Verbindung von AS7018 in Chicago, IL, zur Region USA Ost (Nord-Virginia) herstellen.

```
fields @timestamp,
internetHealth.availability.experienceScore as availabilityExperienceScore,
internetHealth.availability.percentageOfTotalTrafficImpacted as
 percentageOfTotalTrafficImpacted,
internetHealth.performance.experienceScore as performanceExperienceScore,
internetHealth.performance.roundTripTime.p50 as roundTripTimep50,
internetHealth.performance.roundTripTime.p90 as roundTripTimep90,
```

```
internetHealth.performance.roundTripTime.p95 as roundTripTimep95
 | filter clientLocation.country == `United States`
 and clientLocation.city == `Chicago`
 and serviceLocation == `us-east-1`
 and clientLocation.asn == 7018
```

Weitere Informationen finden Sie unter Analysieren von Protokolldaten mit CloudWatch Logs Insights.

Contributor Insights mit Amazon CloudWatch Internet Monitor verwenden

CloudWatch Contributor Insights kann Ihnen dabei helfen, die wichtigsten Kundenstandorte und Netzwerke (ASNs oder Internetdienstanbieter) für Ihre Anwendung zu ermitteln. Verwenden Sie die folgenden Contributor Insights-Beispielregeln, um mit Regeln zu beginnen, die für Amazon CloudWatch Internet Monitor nützlich sind. Weitere Informationen finden Sie unter Eine Contributor-Insights-Regel erstellen.

Weitere Informationen zur Genauigkeit des Standorts von Kunden in Internet Monitor finden Sie unter Geolokalisierungsinformationen und Genauigkeit in Internet Monitor.

> **Note**
>
> Internet Monitor veröffentlicht Daten alle fünf Minuten. Nachdem Sie eine Contributor-Insights-Regel eingerichtet haben, müssen Sie den Zeitraum auf fünf Minuten anpassen, um ein Diagramm zu sehen.

Die wichtigsten Standorte und ASNs anzeigen, die von einer Verfügbarkeitsstörung betroffen sind

Um die wichtigsten Kundenstandorte und ASNs anzuzeigen, die von einem Verfügbarkeitsverlust betroffen sind, können Sie die folgende Contributor-Insights-Regel im Syntax-Editor verwenden. Ersetzen Sie *monitor-name* durch Ihren eigenen Monitornamen.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Sum",
    "Contribution": {
        "Filters": [
            {
```

```
                "Match": "$.clientLocation.city",
                "IsPresent": true
            }
        ],
        "Keys": [
            "$.clientLocation.city",
            "$.clientLocation.networkName"
        ],
        "ValueOf": "$.awsInternetHealth.availability.percentageOfTotalTrafficImpacted"
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "/aws/internet-monitor/monitor-name/byCity"
    ]
}
```

Die wichtigsten Kundenstandorte und ASNs anzeigen, die von Latenzproblemen betroffen sind

Um die wichtigsten Kundenstandorte und ASNs anzuzeigen, die von einem Anstieg der Round-Trip-Zeit (Latenz) betroffen sind, können Sie die folgende Regel von Contributor Insights im Syntax-Editor verwenden. Ersetzen Sie *monitor-name* durch Ihren eigenen Monitornamen.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Sum",
    "Contribution": {
        "Filters": [            {
                "Match": "$.clientLocation.city",
                "IsPresent": true
            }
        ],
        "Keys": [
            "$.clientLocation.city",
            "$.clientLocation.networkName"
        ],
        "ValueOf": "$.awsInternetHealth.performance.percentageOfTotalTrafficImpacted"
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "/aws/internet-monitor/monitor-name/byCity"
```

```
    ]
}
```

Die wichtigsten Kundenstandorte und betroffenen ASNs nach Gesamtanteil am Datenverkehr anzeigen

Um die am stärksten betroffenen Kundenstandorte und ASNs nach dem Gesamtprozentsatz des Datenverkehrs anzuzeigen, können Sie die folgende Regel von Contributor Insights im Syntax-Editor verwenden. Ersetzen Sie *monitor-name* durch Ihren eigenen Monitornamen.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Sum",
    "Contribution": {
        "Filters": [
            {
                "Match": "$.clientLocation.city",
                "IsPresent": true
            }
        ],
        "Keys": [
            "$.clientLocation.city",
            "$.clientLocation.networkName"
        ],
        "ValueOf": "$.percentageOfTotalTraffic"
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "/aws/internet-monitor/monitor-name/byCity"
    ]
}
```

Verwenden von CloudWatch Metriken mit Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor veröffentlicht Kennzahlen für Ihr Konto, einschließlich Messwerte für Leistung, Verfügbarkeit, Round-Trip-Zeit und Durchsatz (Byte pro Sekunde), die Sie in der CloudWatch Konsole unter CloudWatch Metriken einsehen können. Alle Metriken für Ihren Monitor finden Sie im CloudWatch Metrik-Dashboard im benutzerdefinierten Namespace`AWS/InternetMonitor`.

Die Metriken werden für den gesamten Internetverkehr zu Ihren VPCs, Network Load Balancers, CloudFront Distributionen oder WorkSpaces Verzeichnissen im Monitor sowie für den gesamten Datenverkehr zu allen AWS-Region überwachten Internet-Edge-Standorten zusammengefasst. Regionen werden durch den Servicestandort definiert, bei dem es sich entweder um alle Standorte oder um eine bestimmte Region handeln kann, z. B. `us-east-1`.

Hinweis: Stadtnetze sind Kundenstandorte und ASNs (in der Regel Internetdienstanbieter oder ISPs).

Internet Monitor bietet die folgenden Metriken.

Metrik	Beschreibung
PerformanceScore	Eine Leistungsbewertung gibt den geschätzten Prozentsatz des Datenverkehrs an, bei dem kein Leistungsabfall zu verzeichnen ist.
AvailabilityScore	Eine Verfügbarkeitsbewertung stellt den geschätzten Prozentsatz des Datenverkehrs dar, für den kein Verfügbarkeitsrückgang zu verzeichnen ist.
BytesIn	Übertragene Bytes für den Internetverkehr Ihrer Anwendung in allen Stadtnetzen der Anwendung.
BytesOut	Ausgehende übertragene Bytes für den Internetverkehr Ihrer Anwendung in allen Stadtnetzen der Anwendung.
BytesInMonitored	Übertragene Bytes für den Internetverkehr Ihrer Anwendung in überwachten Stadtnetzen.
BytesOutMonitored	Ausgehende übertragene Bytes für den Internetverkehr Ihrer Anwendung in überwachten Stadtnetzen.
Round-Trip-Zeit (RTT)	Die Umlaufzeit zwischen den AWS-Regionen ASNs (in der Regel Internetdienstanbieter oder ISPs) und Standorten (z. B. Städten), die für

Metrik	Beschreibung
	Ihre VPCs, Network Load Balancer, Distribut ionen oder Verzeichnisse spezifisch sind. CloudFront WorkSpaces
CityNetworksMonitored	Die Anzahl der Stadtnetze, die Internet Monitor für den Internetverkehr Ihrer Anwendung überwacht. Dies ist nie mehr als die Höchstgre nze, die Sie als maximale Stadtnetze für den Monitor festgelegt haben.
TrafficMonitoredPercent	Der prozentuale Anteil der Stadtnetze, die von Internet Monitor überwacht werden, am gesamten Anwendungs-Internetverkehr für diesen Monitor. Dieser Wert ist kleiner als 100 (d. h. kleiner als 100 %), wenn Kunden in mehr Stadtnetzen auf Ihre Anwendung zugreifen als die maximale Anzahl von Stadtnetzen, die Sie für den Monitor festgelegt haben.
CityNetworksFor100 PercentTraffic	Die Zahl, auf die Sie die Höchstgrenze Ihres Stadtnetzes setzen sollten, wenn Sie 100 % des Internetverkehrs Ihrer Anwendung in Internet Monitor überwachen möchten.
CityNetworksFor99 PercentTraffic	Die Zahl, auf die Sie die Höchstgrenze Ihres Stadtnetzes setzen sollten, wenn Sie 99 % des Internetverkehrs Ihrer Anwendung in Internet Monitor überwachen möchten.
CityNetworksFor95 PercentTraffic	Die Zahl, auf die Sie die Höchstgrenze Ihres Stadtnetzes setzen sollten, wenn Sie 95 % des Internetverkehrs Ihrer Anwendung in Internet Monitor überwachen möchten.

Metrik	Beschreibung
CityNetworksFor90 PercentTraffic	Die Zahl, auf die Sie die Höchstgrenze Ihres Stadtnetzes setzen sollten, wenn Sie 90 % des Internetverkehrs Ihrer Anwendung in Internet Monitor überwachen möchten.
CityNetworksFor75 PercentTraffic	Die Zahl, auf die Sie die Höchstgrenze Ihres Stadtnetzes setzen sollten, wenn Sie 75 % des Internetverkehrs Ihrer Anwendung in Internet Monitor überwachen möchten.
CityNetworksFor50 PercentTraffic	Die Zahl, auf die Sie die Höchstgrenze Ihres Stadtnetzes setzen sollten, wenn Sie 50 % des Internetverkehrs Ihrer Anwendung in Internet Monitor überwachen möchten.
CityNetworksFor25 PercentTraffic	Die Zahl, auf die Sie die Höchstgrenze Ihres Stadtnetzes setzen sollten, wenn Sie 25 % des Internetverkehrs Ihrer Anwendung in Internet Monitor überwachen möchten.

> **Note**
>
> Beispiele für die Verwendung mehrerer dieser Metriken zur Bestimmung der Werte für die Auswahl eines Höchstwerts für Stadtnetze für Ihren Monitor finden Sie unter Auswählen des Maximalwerts für Stadtnetze.

Weitere Informationen finden Sie unter Verwenden Sie CloudWatch Amazon-Metriken.

Verwendung von Amazon Athena zur Abfrage von Internetmessungen in Amazon-S3-Protokolldateien

Sie können Amazon Athena verwenden, um die Internet-Messwerte, die Amazon CloudWatch Internet Monitor in einem Amazon S3 S3-Bucket veröffentlicht, abzufragen und anzuzeigen. Es gibt eine Option in Internet Monitor, mit der Sie Internetmessungen für Ihre Anwendung in einem S3-

Bucket für den internetbezogenen Datenverkehr für Ihre überwachten Stadtnetze (Kundenstandorte und ASNs, in der Regel Internetdienstanbieter oder ISPs) veröffentlichen können. Unabhängig davon, ob Sie Messungen auf S3 veröffentlichen, veröffentlicht Internet Monitor automatisch alle fünf Minuten Internet-Messungen in CloudWatch Logs für die 500 größten Stadtnetzwerke (nach Verkehrsaufkommen) für jeden Monitor.

In diesem Kapitel erfahren Sie, wie Sie in Athena eine Tabelle für Internetmessungen in einer S3-Protokolldatei erstellen. Anschließend finden Sie Beispielabfragen, um verschiedene Ansichten der Messungen zu sehen. Sie können beispielsweise nach Ihren 10 am stärksten betroffenen Stadtnetze nach Latenzauswirkungen abfragen.

Verwenden von Amazon Athena zum Erstellen einer Tabelle für Internetmessungen in Internet Monitor

Um Athena mit Ihren S3-Protokolldateien von Internet Monitor zu verwenden, erstellen Sie zunächst eine Tabelle für die Internetmessungen.

Folgen Sie den Schritten in diesem Verfahren, um eine Tabelle in Athena auf der Grundlage der S3-Protokolldateien zu erstellen. Anschließend können Sie Athena-Abfragen für die Tabelle ausführen, z. B. diese Beispielabfragen für Internetmessungen, um Informationen über Ihre Messungen zu erhalten.

So erstellen Sie eine Athena-Tabelle

1. Öffnen Sie die Athena-Konsole unter https://console.aws.amazon.com/athena/.
2. Geben Sie im Athena-Abfrage-Editor eine Abfrageanweisung ein, um eine Tabelle mit Internetmessungen von Internet Monitor zu erstellen. Ersetzen Sie den Wert für den Parameter LOCATION durch den Speicherort des S3-Buckets, in dem Ihre Internetmessungen von Internet Monitor gespeichert sind.

```
CREATE EXTERNAL TABLE internet_measurements (
    version INT,
    timestamp INT,
    clientlocation STRING,
    servicelocation STRING,
    percentageoftotaltraffic DOUBLE,
    bytesin INT,
    bytesout INT,
    clientconnectioncount INT,
    internethealth STRING,
```

```
    trafficinsights STRING
)
PARTITIONED BY (year STRING, month STRING, day STRING)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION
's3://bucket_name/bucket_prefix/AWSLogs/account_id/internetmonitor/AWS_Region/'
TBLPROPERTIES ('skip.header.line.count' = '1');
```

3. Geben Sie eine Anweisung ein, um eine Partition zum Lesen der Daten zu erstellen. Die folgende Abfrage erstellt zum Beispiel eine einzelne Partition für ein bestimmtes Datum und einen bestimmten Ort:

```
ALTER TABLE internet_measurements
ADD PARTITION (year = 'YYYY', month = 'MM', day = 'dd')
LOCATION
's3://bucket_name/bucket_prefix/AWSLogs/account_id/internetmonitor/AWS_Region/YYYY/
MM/DD';
```

4. Wählen Sie Ausführen aus.

Athena-Beispielanweisungen für Internetmessungen

Im Folgenden finden Sie ein Beispiel für eine Anweisung zur Erstellung einer Tabelle:

```
CREATE EXTERNAL TABLE internet_measurements (
    version INT,
    timestamp INT,
    clientlocation STRING,
    servicelocation STRING,
    percentageoftotaltraffic DOUBLE,
    bytesin INT,
    bytesout INT,
    clientconnectioncount INT,
    internethealth STRING,
    trafficinsights STRING
)
PARTITIONED BY (year STRING, month STRING, day STRING)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION 's3://internet-measurements/TestMonitor/AWSLogs/1111222233332/internetmonitor/
us-east-2/'
TBLPROPERTIES ('skip.header.line.count' = '1');
```

Im Folgenden finden Sie ein Beispiel für eine Anweisung zum Erstellen einer Partition zum Lesen der Daten:

```
ALTER TABLE internet_measurements
ADD PARTITION (year = '2023', month = '04', day = '07')
LOCATION 's3://internet-measurements/TestMonitor/AWSLogs/1111222233332/internetmonitor/
us-east-2/2023/04/07/'
```

Beispiele für Amazon-Athena-Abfragen zur Verwendung mit Internetmessungen in Internet Monitor

Dieser Abschnitt enthält Beispielabfragen, die Sie mit Amazon Athena verwenden können, um Informationen über die in Amazon S3 veröffentlichten Internetmessungen Ihrer Anwendung zu erhalten.

Fragen Sie Ihre 10 am häufigsten betroffenen Kundenstandorte und ASNs ab (gemessen am Gesamtanteil des Verkehrs)

Führen Sie diese Athena-Abfrage aus, um die 10 am stärksten betroffenen Stadtnetze (nach Gesamtprozentsatz des Datenverkehrs) zu ermitteln, d. h. Kundenstandorte und ASNs, in der Regel Internetdienstanbieter.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.networkname') as networkName,
    sum(percentageoftotaltraffic) as percentageoftotaltraffic
FROM internet_measurements
GROUP BY json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.networkname')
ORDER BY percentageoftotaltraffic desc
limit 10
```

Fragen Sie Ihre 10 (nach Verfügbarkeit) am häufigsten betroffenen Kundenstandorte und ASNs ab

Führen Sie diese Athena-Abfrage aus, um die 10 am stärksten betroffenen Stadtnetze (nach Gesamtprozentsatz des Datenverkehrs) zu ermitteln, d. h. Kundenstandorte und ASNs, in der Regel Internetdienstanbieter.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.networkname') as networkName,
    sum(
        cast(
            json_extract_scalar(
                internetHealth,
```

```
                '$.availability.percentageoftotaltrafficimpacted'
            )
        as double )
    ) as percentageOfTotalTrafficImpacted
FROM internet_measurements
GROUP BY json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.networkname')
ORDER BY percentageOfTotalTrafficImpacted desc
limit 10
```

Ihre 10 (nach Latenz) am stärksten betroffenen Kundenstandorte und ASNs abfragen

Führen Sie diese Athena-Abfrage aus, um die 10 Stadtnetze zu ermitteln, die am stärksten von der Latenz betroffen sind, d. h. Kundenstandorte und ASNs, in der Regel Internetdienstanbieter.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.networkname') as networkName,
    sum(
        cast(
            json_extract_scalar(
                internetHealth,
                '$.performance.percentageoftotaltrafficimpacted'
            )
        as double )
    ) as percentageOfTotalTrafficImpacted
FROM internet_measurements
GROUP BY json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.networkname')
ORDER BY percentageOfTotalTrafficImpacted desc
limit 10
```

Die Datenverkehrs-Highlights für Ihre Kundenstandorte und ASNs abfragen

Führen Sie diese Athena-Abfrage aus, um für Ihre Stadtnetze – d. h. Kundenstandorte und ASNs, in der Regel Internetdienstanbieter – Highlights zum Datenverkehr zu erhalten, einschließlich Verfügbarkeitsbewertung, Leistungsbewertung und Zeit bis zum ersten Byte.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.subdivision') as subdivision,
    json_extract_scalar(clientLocation, '$.country') as country,
    avg(cast(json_extract_scalar(internetHealth, '$.availability.experiencescore') as
 double)) as availabilityScore,
```

```
    avg(cast(json_extract_scalar(internetHealth, '$.performance.experiencescore') as
 double)) performanceScore,
    avg(cast(json_extract_scalar(trafficinsights,
 '$.timetofirstbyte.currentexperience.value') as double)) as averageTTFB,
    sum(bytesIn) as bytesIn,
    sum(bytesOut) as bytesOut,
    sum(bytesIn + bytesOut) as totalBytes
FROM internet_measurements
where json_extract_scalar(clientLocation, '$.city') != 'N/A'
GROUP BY
json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.subdivision'),
    json_extract_scalar(clientLocation, '$.country')
ORDER BY totalBytes desc
limit 100
```

Weitere Informationen zur Verwendung von Athena finden Sie im Amazon Athena-Benutzerhandbuch.

Verwenden der Amazon CloudWatch Internet Monitor-Abfrageschnittstelle

Eine Option, um mehr über den Internetverkehr für Ihre AWS Anwendung zu erfahren, ist die Verwendung der Amazon CloudWatch Internet Monitor-Abfrageschnittstelle. Um die Abfrageschnittstelle zu verwenden, erstellen Sie eine Abfrage mit von Ihnen ausgewählten Datenfiltern und führen dann die Abfrage aus, um eine Teilmenge Ihrer Internet-Monitor-Daten zurückzugeben. Wenn Sie die Daten untersuchen, die die Abfrage zurückgibt, erhalten Sie Einblicke in die Leistung Ihrer Anwendung im Internet.

Sie können alle Messwerte, die Internet Monitor mit Ihrem Monitor erfasst, abfragen und untersuchen, darunter Verfügbarkeits- und Leistungswerte, übertragene Byte, Round-Trip-Zeiten und Time to First Byte (TTFB).

Internet Monitor verwendet die Abfrageschnittstelle, um die Daten bereitzustellen, die Sie im Dashboard der Internet-Monitor-Konsole untersuchen können. Mithilfe der Suchoptionen im Dashboard – auf der Registerkarte Historischer Entdecker oder der Registerkarte Datenverkehreinblicke – können Sie Internetdaten für Ihre Anwendung abfragen und filtern.

Wenn Sie mehr Flexibilität beim Durchsuchen und Filtern Ihrer Daten wünschen, als das Dashboard bietet, können Sie die Abfrageschnittstelle selbst verwenden, indem Sie Internet Monitor-API-Operationen mit dem AWS Command Line Interface oder mit einem AWS SDK verwenden. In diesem Abschnitt werden die Abfragetypen vorgestellt, die Sie mit der Abfrageschnittstelle verwenden

können, und die Filter, die Sie angeben können, um eine Teilmenge von Daten zu erstellen, um Einblicke in den Internetverkehr für Ihre Anwendung zu erhalten.

Themen

So benutzt man die Abfrageschnittstelle

Sie erstellen eine Abfrage mit der Abfrageschnittstelle, indem Sie einen Abfragetyp auswählen und dann Filterwerte angeben, um eine bestimmte gewünschte Teilmenge Ihrer Protokolldateidaten zurückzugeben. Anschließend können Sie mit der Datenteilmenge arbeiten, um weiter zu filtern und zu sortieren, Berichte zu erstellen usw.

Der Abfrageprozess funktioniert wie folgt:

1. Wenn Sie eine Abfrage ausführen, gibt Internet Monitor eine `query ID` zurück, der für die Abfrage eindeutig ist. In diesem Abschnitt werden die verfügbaren Abfragetypen und Optionen zum Filtern von Daten in Abfragen beschrieben. Um zu verstehen, wie das funktioniert, können Sie sich auch den Abschnitt mit Abfragebeispielen ansehen.
2. Bei der GetQueryResultsAPI-Operation, um Datenergebnisse für die Abfrage zurückzugeben, geben Sie die Abfrage-ID zusammen mit Ihrem Monitornamen an. Jeder Abfragetyp gibt einen anderen Satz von Datenfeldern zurück. Weitere Informationen finden Sie unter Abfrageergebnisse abrufen.

Die Abfrageschnittstelle bietet die folgenden drei Abfragetypen. Jeder Abfragetyp gibt einen anderen Satz von Informationen über Ihren Datenverkehr aus den Protokolldateien zurück, wie in der Abbildung gezeigt.

- Messungen: Zeigt die Verfügbarkeitsbewertung, die Leistungsbewertung, den Gesamtverkehr und die Hin- und Rückflugzeiten in Intervallen von 5 Minuten an.
- Top-Standorte: Stellt Verfügbarkeitsbewertung, Leistungsbewertung, Gesamtdatenverkehr und TTFB-Informationen (Time to First Byte) für die wichtigsten Standort- und ASN-Kombinationen, die Sie überwachen, aufgeschlüsselt nach Verkehrsaufkommen bereit.

- Details zu den wichtigsten Standorten: Stellt TTFB für Amazon CloudFront, Ihre aktuelle Konfiguration und die leistungsstärkste Amazon EC2 EC2-Konfiguration in Intervallen von 1 Stunde bereit.

Mit jedem dieser Abfragetypen können Sie die Daten weiter filtern, indem Sie eines oder mehrere der folgenden Kriterien angeben:

- AWS Standort: Als AWS Standort können Sie einen CloudFront oder einen angeben AWS-Region`us-east-2`, z. B.`us-west-2`, usw.
- ASN: Geben Sie einen ASN an, bei dem es sich in der Regel um einen Internetdienstanbieter (ISP) handelt.
- Kundenstandort: Geben Sie als Standort eine Stadt, eine Metro-Region, eine Unterteilung oder ein Land an.
- Geo: Geben Sie `geo` für einige Abfragen an. Dies ist für Abfragen erforderlich, die den `Top locations`-Abfragetyp verwenden, ist aber für andere Abfragetypen nicht zulässig. Informationen darüber, wann `geo` für Filterparameter angegeben werden muss, finden Sie im Abschnitt mit Abfragebeispielen.

Die Operatoren, die Sie zum Filtern Ihrer Daten verwenden können, sind `EQUALS` und `NOT_EQUALS`. Einzelheiten zum Filtern von Parametern finden Sie unter FilterParameterAPI-Vorgang.

Einzelheiten zu den Vorgängen der Abfrageschnittstelle finden Sie in den folgenden API-Operationen im Amazon CloudWatch Internet Monitor API-Referenzhandbuch:

- Informationen zum Erstellen und Ausführen einer Abfrage finden Sie unter StartQueryAPI-Vorgang.
- Informationen zum Beenden einer Abfrage finden Sie unter StopQueryAPI-Vorgang.
- Informationen zum Zurückgeben von Daten für eine von Ihnen erstellte Abfrage finden Sie unter GetQueryResultsAPI-Vorgang.
- Informationen zum Abrufen des Status einer Abfrage finden Sie unter GetQueryStatusAPI-Vorgang.

Abfragebeispiele

Um eine Abfrage zu erstellen, mit der Sie einen gefilterten Datensatz aus der Protokolldatei Ihres Monitors abrufen können, verwenden Sie den StartQueryAPI-Vorgang. Sie geben einen Abfragetyp und Filterparameter für die Abfrage an. Anschließend, wenn Sie den API-Vorgang der Internet-

Monitor-Abfrageschnittstelle verwenden, um mithilfe der Abfrage Ergebnisse zu erhalten, wird die Teilmenge Ihrer Daten abgerufen, mit der Sie arbeiten möchten.

Schauen wir uns einige Beispiele an, um zu veranschaulichen, wie Abfragetypen und Filterparameter funktionieren.

Beispiel 1

Nehmen wir an, Sie möchten alle Protokolldateidaten Ihres Monitors für ein bestimmtes Land abrufen, mit Ausnahme einer Stadt. Das folgende Beispiel zeigt Filterparameter für eine Abfrage, die Sie mit dem Vorgang `StartQuery` für dieses Szenario erstellen könnten.

```
{
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "MEASUREMENTS"
   FilterParameters: [
      {
       Field: "country",
       Operator: "EQUALS",
       Values: ["Germany"]
      },
      {
       Field: "city",
       Operator: "NOT_EQUALS",
       Values: ["Berlin"]
      },
    ]
}
```

Beispiel 2

Als weiteres Beispiel wird angenommen, Sie möchten Ihre Top-Standorte nach Metropolregion gefiltert anzeigen. Sie könnten die folgende Beispielabfrage für dieses Szenario verwenden.

```
{
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "TOP_LOCATIONS"
   FilterParameters: [
```

```
      {
       Field: "geo",
       Operator: "EQUALS",
       Values: ["metro"]
      },
    ]
}
```

Beispiel 3

Nehmen wir nun an, Sie möchten sich die Top-Kombinationen aus Städten und Netzwerken in der Metropolregion Los Angeles ansehen. Geben Sie dazu `geo=city` an und setzen Sie `metro` auf Los Angeles. Jetzt gibt die Abfrage die Top-Städtenetze in der Metropolregion Los Angeles zurück anstatt der Top-Ergebnisse für Metro+-Netzwerke insgesamt.

Hier ist die Beispielabfrage, die Sie verwenden könnten:

```
{
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "TOP_LOCATIONS"
   FilterParameters: [
      {
       Field: "geo",
       Operator: "EQUALS",
       Values: ["city"]
      },
      {
       Field: "metro",
       Operator: "EQUALS",
       Values: ["Los Angeles"]
      }
    ]
}
```

Beispiel 4

Nehmen wir abschließend an, dass Sie TTFB-Daten für eine bestimmte Unterteilung (z. B. einen US-Bundesstaat) abrufen möchten.

Im Folgenden finden Sie eine Beispielabfrage für dieses Szenario:

```
{
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "TOP_LOCATION_DETAILS"
   FilterParameters: [
      {
       Field: "subdivision",
       Operator: "EQUALS",
       Values: ["California"]
      },
    ]
}
```

Abfrageergebnisse abrufen

Nachdem Sie eine Abfrage definiert haben, können Sie mit der Abfrage eine Reihe von Ergebnissen zurückgeben, indem Sie einen weiteren Internet Monitor-API-Vorgang ausführen GetQueryResults. Bei der Ausführung von `GetQueryResults` geben Sie die Abfrage-ID für die von Ihnen definierte Abfrage zusammen mit dem Namen Ihres Monitors an. `GetQueryResults` ruft Daten für die angegebene Abfrage in einen Ergebnissatz ab.

Vergewissern Sie sich beim Ausführen einer Abfrage, dass die Ausführung der Abfrage abgeschlossen ist, bevor Sie `GetQueryResults` verwenden, um die Ergebnisse anzusehen. Mithilfe des GetQueryStatusAPI-Vorgangs können Sie feststellen, ob die Abfrage abgeschlossen wurde. Wenn der `Status` für die Abfrage `SUCCEEDED` ist, können Sie mit der Überprüfung der Ergebnisse fortfahren.

Wenn Ihre Abfrage abgeschlossen ist, können Sie die folgenden Informationen verwenden, um die Ergebnisse zu überprüfen. Jeder Abfragetyp, den Sie zum Erstellen einer Abfrage verwenden, enthält einen eindeutigen Satz von Datenfeldern aus den Protokolldateien, wie in der folgenden Liste beschrieben:

Messungen

Der Abfragetyp `measurements` gibt die folgenden Daten zurück:

```
timestamp, availability, performance, bytes_in, bytes_out, rtt_p50,
rtt_p90, rtt_p95
```

Top-Standorte

Der Abfragetyp `top locations` gruppiert Daten nach Standort und stellt die über den Zeitraum gemittelten Daten bereit. Die zurückgegebenen Daten umfassen Folgendes:

`aws_location, city, metro, subdivision, country, asn, availability, performance, bytes_in, bytes_out, current_fbl, best_ec2, best_ec2_region, best_cf_fbl`

Beachten Sie, dass `city`, `metro` und `subdivision` nur zurückgegeben werden, wenn Sie diesen Standorttyp für das `geo`-Feld wählen. Je nach dem Standorttyp, den Sie angeben, werden die folgenden Standortfelder für `geo` zurückgegeben:

```
city = city, metro, subdivision, country
metro = metro, subdivision, country
subdivision = subdivision, country
country = country
```

Top-Standort-Details

Der Abfragetyp `top locations details` gibt Daten nach Stunden gruppiert zurück. Die Abfrage gibt die folgenden Daten zurück:

`timestamp, current_service, current_fbl, best_ec2_fbl, best_ec2_region, best_cf_fbl`

Wenn Sie den API-Vorgang `GetQueryResults` ausführen, gibt Internet Monitor in der Antwort Folgendes zurück:

- Ein Datenzeichenfolge-Array, das die Ergebnisse enthält, die die Abfrage zurückgibt. Die Informationen werden in Arrays zurückgegeben, die auf das Feld `Fields` ausgerichtet sind. Dies wird auch durch den API-Aufruf zurückgegeben. Mithilfe des `Fields`-Felds können Sie die Informationen aus dem `Data`-Repository analysieren und sie dann für Ihre Zwecke weiter filtern oder sortieren.
- Ein Array von Feldern, das die Felder auflistet, für die die Abfrage Daten zurückgegeben hat (in der `Data`-Feldantwort). Jedes Element im Array ist ein Name-Datentyp-Paar, z. B. `availability_score-float`.

Fehlerbehebung

Wenn bei der Verwendung von API-Operationen für die Abfrageschnittstelle Fehler zurückgegeben werden, stellen Sie sicher, dass Sie über die erforderlichen Berechtigungen zur Verwendung von Amazon CloudWatch Internet Monitor verfügen. Stellen Sie insbesondere sicher, dass Sie über die folgenden Berechtigungen verfügen:

```
internetmonitor:StartQuery
internetmonitor:GetQueryStatus
internetmonitor:GetQueryResults
internetmonitor:StopQuery
```

Diese Berechtigungen sind in der empfohlenen AWS Identity and Access Management Richtlinie für die Verwendung des Internet Monitor-Dashboards in der Konsole enthalten. Weitere Informationen finden Sie unter IAM-Berechtigungen für Amazon CloudWatch Internet Monitor.

Alarme mit Amazon CloudWatch Internet Monitor erstellen

Sie können CloudWatch Amazon-Alarme auf der Grundlage von Amazon CloudWatch Internet Monitor-Metriken erstellen, genau wie Sie es für andere CloudWatch Amazon-Metriken tun können.

Sie können z. B. einen Alarm auf der Grundlage der Internet-Monitor-Metrik `PerformanceScore` erstellen und ihn so konfigurieren, dass er eine Benachrichtigung sendet, wenn die Metrik unter einem von Ihnen gewählten Wert liegt. Sie konfigurieren Alarme für Internet Monitor-Metriken nach denselben Richtlinien wie für andere CloudWatch Messwerte.

Im Folgenden finden Sie Beispiele für Internet-Monitor-Metriken, für die Sie möglicherweise einen Alarm erstellen möchten:

- PerformanceScore
- AvailabilityScore
- RoundtripTime

Alle für Internet Monitor verfügbaren Metriken finden Sie unter Verwenden von CloudWatch Metriken mit Amazon CloudWatch Internet Monitor.

Das folgende Verfahren bietet ein Beispiel für die Aktivierung eines Alarms, PerformanceScoreindem Sie im CloudWatch Dashboard zu der Metrik navigieren. Anschließend folgen Sie den CloudWatch

Standardschritten, um einen Alarm auf der Grundlage eines von Ihnen ausgewählten Schwellenwerts zu erstellen und eine Benachrichtigung einzurichten oder andere Optionen auszuwählen.

Um einen Alarm für PerformanceScorein CloudWatch Metrics zu erstellen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie Metriken und dann Alle Metriken.
3. Filtern Sie nach Internet Monitor, indem Sie `AWS/InternetMonitor` wählen.
4. Wähle MeasurementSource, MonitorName.
5. Wählen Sie in der Liste aus PerformanceScore.
6. Wählen Sie auf der GraphedMetricsRegisterkarte unter Aktionen das Glockensymbol aus, um einen Alarm auf der Grundlage eines statischen Schwellenwerts zu erstellen.

Folgen Sie nun den CloudWatch Standardschritten, um Optionen für den Alarm auszuwählen. Sie können sich beispielsweise dafür entscheiden, per Amazon SNS SNS-Nachricht benachrichtigt zu werden, wenn ein bestimmter Schwellenwert unterschritten PerformanceScorewird. Alternativ oder zusätzlich können Sie den Alarm zu einem Dashboard hinzufügen.

Beachten Sie Folgendes:

- Internet-Monitor-Metriken werden in der Regel innerhalb von 20 Minuten berechnet und veröffentlicht.
- Wenn Sie einen Alarm auf der Grundlage von Internet-Monitor-Metriken erstellen, stellen Sie sicher, dass Sie bei der Festlegung des Lookback-Zeitraums für einen Alarm die kurze Verzögerung vor der Veröffentlichung berücksichtigen. Es wird empfohlen, Auswertungszeiträume mit einem Lookback-Zeitraum von mindestens 25 Minuten zu konfigurieren.

Weitere Informationen zur Verwendung von CloudWatch Alarmen mit Internet Monitor finden Sie im folgenden Blogbeitrag: Verwenden von Amazon CloudWatch Internet Monitor für verbesserte Internetbeobachtbarkeit.

Weitere Informationen zu den Optionen beim Erstellen eines CloudWatch Alarms finden Sie unterErstellen Sie einen CloudWatch Alarm auf der Grundlage eines statischen Schwellenwerts.

Amazon CloudWatch Internet Monitor mit Amazon verwenden EventBridge

Die Integritätsereignisse, die Amazon CloudWatch Internet Monitor für Netzwerkprobleme erstellt EventBridge, werden bei Amazon veröffentlicht, sodass Sie Benachrichtigungen über jede Verschlechterung der Benutzererfahrung für Ihre Anwendung senden können.

Folgen Sie der Anleitung hier, EventBridge um mit Internet Monitor-Gesundheitsereignissen zu arbeiten.

So richten Sie eine Regel für Internet Monitor in ein EventBridge

1. Wählen Sie im AWS Management Console, in EventBridge, Regeln aus und geben Sie dann einen Namen und eine Beschreibung ein. Erstellen Sie die Regel auf dem Default (Standard)-Event-Bus.
2. Wählen Sie in Schritt 2 Andere als Ereignisquelle aus, und ordnen Sie dann unter Ereignismuster die folgende Quelle zu.

   ```
   {
     "source": ["aws.internetmonitor"]
   }
   ```
3. Wählen Sie in Schritt 3 als Ziel die Option AWS Service and CloudWatch Logs Group aus und wählen Sie dann eine bestehende Protokollgruppe aus, oder erstellen Sie eine neue.
4. Fügen Sie alle gewünschten Tags hinzu und erstellen Sie dann die Regel. Dadurch sollte Ihre ausgewählte CloudWatch Protokollgruppe mit Ereignissen von EventBridge gefüllt werden.

Weitere Informationen darüber, wie EventBridge Regeln mit Ereignismustern funktionieren, finden Sie unter EventBridge Amazon-Ereignismuster im EventBridge Amazon-Benutzerhandbuch.

Beheben Sie Fehler beim Zugriff auf CloudWatch Protokolle und Metriken

Um einige Funktionen zu unterstützen, muss Amazon CloudWatch Internet Monitor mit bestimmten CloudWatch Amazon-Ressourcen interagieren, einschließlich Protokollen und Metriken. Wenn Internet Monitor nicht auf die CloudWatch Ressourcen zugreifen kann, auf die er Zugriff benötigt, legt Internet Monitor den Statuscode `FAULT_ACCESS_CLOUDWATCH` für den Monitor fest.

Es gibt mehrere Gründe, warum Ihr Monitor den Status `FAULT_ACCESS_CLOUDWATCH` haben könnte. In den folgenden Abschnitten werden mögliche Ursachen für diese Fehler sowie Vorschläge zur Problembehandlung aufgeführt.

Internet Monitor konnte nicht auf die CloudWatch Protokolle in Ihrem Konto zugreifen

Internet Monitor veröffentlicht Diagnoseprotokolle über den Anwendungsdatenverkehr, den Ihr Monitor verfolgt. Es veröffentlicht diese Protokolle in Protokollgruppen in CloudWatch Protokollen am folgenden Speicherort:`/aws/internet-monitor/`*`monitor_name`*`/[byCity|byMetro|bySubdivision|byCountry]`. Internet Monitor konnte nicht auf diese Protokollgruppen zugreifen.

Fehlerstatus und mögliche Lösungen:

- PutLogEvents Drosselungsfehler: Der Internet Monitor-Dienst wurde möglicherweise gedrosselt, als er versuchte, die Protokolle Ihres Monitors auf zu veröffentlichen. CloudWatch Überprüfen Sie die Drosselungslimits für Ihr Konto und fordern Sie gegebenenfalls eine Erhöhung des Limits an.
- Protokollgruppe nicht gefunden: Deaktivieren Sie Ihren Monitor, und aktivieren Sie ihn anschließend erneut. Durch die Aktivierung eines Monitors wird die Erstellung der Protokollgruppe neu gestartet, wodurch das Problem möglicherweise behoben wird.
- PutLogEvents Fehler „Zugriff verweigert“: Wenden Sie sich an den AWS Support, um Unterstützung zu erhalten.
- PutLogEvents unbekannter oder allgemeiner Fehler: Wenden Sie sich an den AWS Support, um Unterstützung zu erhalten.

Internet Monitor konnte nicht auf die CloudWatch Messwerte in Ihrem Konto zugreifen

Internet Monitor liefert spezifische CloudWatch Messwerte zum Anwendungsdatenverkehr, der von einem Monitor verfolgt wird. Beim Versuch von Internet Monitor, diese Messwerte an zu übermitteln, ist ein Fehler aufgetreten CloudWatch.

Fehlerstatus und mögliche Lösungen:

- PutMetricData Drosselungsfehler: Der Internet Monitor-Dienst wurde möglicherweise gedrosselt, als er versuchte, die Messwerte Ihres Monitors auf zu veröffentlichen. CloudWatch Überprüfen Sie die Drosselungslimits für Ihr Konto und fordern Sie gegebenenfalls eine Erhöhung des Limits an.
- PutMetricData Fehler „Zugriff verweigert“: Wenden Sie AWS sich an den Support, um Unterstützung zu erhalten.
- PutMetricData unbekannter oder allgemeiner Fehler: Wenden Sie sich an den AWS Support, um Unterstützung zu erhalten.

Datenschutz und Datenschutz mit Amazon CloudWatch Internet Monitor

Das Modell der AWS gemeinsamen Verantwortung gilt für den Datenschutz und den Datenschutz in Amazon CloudWatch Internet Monitor. AWS Ist, wie in diesem Modell beschrieben, für den Schutz der globalen Infrastruktur verantwortlich, auf der die gesamte AWS Cloud betrieben wird. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blogbeitrag The AWS Shared Responsibility Model und GDPR im AWS Security Blog. Weitere Ressourcen zur Einhaltung der DSGVO-Anforderungen finden Sie im Center für Datenschutz-Grundverordnung (DSGVO).

Wir empfehlen dringend, in Freitextfeldern keine sensiblen personenbezogenen Daten wie Kontonummern von Endkunden, E-Mail-Adressen oder sonstige personenbezogenen Daten einzugeben. Alle Daten, die Sie in Amazon CloudWatch Internet Monitor oder andere Dienste eingeben, können in Diagnoseprotokollen enthalten sein.

Identity and Access Management für Amazon CloudWatch Internet Monitor

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf AWS Ressourcen sicher zu kontrollieren. IAM-Administratoren steuern, wer für die Nutzung von Internet-Monitor-Ressourcen authentifiziert (angemeldet) und autorisiert (mit Berechtigungen ausgestattet) werden kann. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Important

Änderungen der Internet-Monitor-Ressourcen am 24. Februar 2023
Wenn Sie vor dem 24. Februar 2023 IAM-Richtlinien erstellt haben, die Internet-Monitor-Ressourcen enthalten, beachten Sie die folgenden Änderungen an Internet-Monitor-Ressourcen und -Ressourcentypen.

- HealthEventsRessource wurde umbenannt in. HealthEvent
- Die ARN- und Regex-Formate für die HealthEventRessource wurden aktualisiert.
- Die ARN- und Regex-Formate für die Monitor-Ressource wurden aktualisiert.
- Berechtigungen auf Ressourcenebene für die GetHealthEventAktion werden jetzt nur für den Ressourcentyp unterstützt. HealthEvent Sie werden auf der Monitor-Ressource nicht unterstützt.

- TagResource, UntagResource, und ListTagsForResourcefür den Ressourcentyp Monitor wurden aktualisiert und sind nun erforderlich.

Weitere Informationen zu den Aktionen, Ressourcen und Bedingungsschlüsseln, die Sie in Richtlinien zur Verwaltung des Zugriffs auf AWS Ressourcen in Internet Monitor angeben können, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon CloudWatch Internet Monitor.

Inhalt

So funktioniert Amazon CloudWatch Internet Monitor mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf Internet Monitor zu verwalten, erfahren Sie, welche IAM-Features für Internet Monitor zur Verfügung stehen.

Tabellen, die einen ähnlichen Überblick darüber bieten, wie AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren.

IAM-Funktionen, die Sie mit Amazon CloudWatch Internet Monitor verwenden können

IAM-Feature	Unterstützung für Internet Monitor
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja

IAM-Feature	Unterstützung für Internet Monitor
Richtlinienbedingungsschlüssel (servicespezifisch)	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Teilweise
Temporäre Anmeldeinformationen	Ja
Hauptberechtigungen	Ja
Servicerollen	Nein
Serviceverknüpfte Rollen	Ja

Identitätsbasierte Richtlinien für Internet Monitor

Unterstützt Richtlinien auf Identitätsbasis.	Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien in Internet Monitor

Unterstützt ressourcenbasierte Richtlinien	Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern.

Richtlinienaktionen für Internet Monitor

Unterstützt Richtlinienaktionen	Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der Internet Monitor-Aktionen finden Sie unter Von Amazon CloudWatch Internet Monitor definierte Aktionen in der Service Authorization Reference.

Richtlinienaktionen in Internet Monitor verwenden das folgende Präfix vor der Aktion:

```
internetmonitor
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
      "internetmonitor:action1",
      "internetmonitor:action2"
         ]
```

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

```
"Action": "internetmonitor:Describe*"
```

Richtlinienressourcen für Internet Monitor

Unterstützt Richtlinienressourcen	Ja

In der Referenz für die Service-Autorisierung finden Sie die folgenden Informationen zu Internet Monitor:

- Eine Liste der Internet Monitor-Ressourcentypen und ihrer ARNs finden Sie unter Von Amazon CloudWatch Internet Monitor definierte Ressourcen.
- Informationen zu den Aktionen, die Sie mit dem ARN jeder Ressource angeben können, finden Sie unter Von Amazon CloudWatch Internet Monitor definierte Aktionen.

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein – `Resource`oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Richtlinien-Bedingungsschlüssel für Internet Monitor

Unterstützt servicespezifische Richtlinienbedingungsschlüssel	Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen `AND`-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen `OR` Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

Eine Liste der Internet Monitor-Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für Amazon CloudWatch Internet Monitor in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von Amazon CloudWatch Internet Monitor definierte Aktionen.

ACLs in Internet Monitor

Unterstützt ACLs	Nein

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit Internetmonitor

Unterstützt ABAC (Tags in Richtlinien)	Teilweise

Internet Monitor unterstützt teilweise Tags in Richtlinien. Es unterstützt das Tagging für eine Ressource, Monitore.

Um Tags mit Internet Monitor zu verwenden, verwenden Sie das AWS Command Line Interface oder ein AWS SDK. Tagging für Internet Monitor wird mit dem AWS Management Console nicht unterstützt.

Wenn Sie mehr über die Verwendung von Tags in Richtlinien im Allgemeinen erfahren möchten, lesen Sie die folgenden Informationen.

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In AWS werden diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Was ist ABAC? im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

Verwenden von temporären Anmeldeinformationen mit Internet Monitor

Unterstützt temporäre Anmeldeinformationen	Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services funktionieren AWS-Services , finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter Wechseln zu einer Rolle (Konsole) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldeinformationen in IAM.

Serviceübergreifende Prinzipal-Berechtigungen für Internet Monitor

Unterstützt Forward Access Sessions (FAS)	Ja

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für Internet Monitor

Unterstützt Servicerollen	Nein

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

Serviceverknüpfte Rolle für Internet Monitor

Unterstützt serviceverknüpfte Rollen	Ja

Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Weitere Informationen über die serviceverknüpfte Rolle für Internet Monitor finden Sie unter Servicebezogene Rolle für Amazon CloudWatch Internet Monitor.

Weitere Informationen zum Erstellen oder Verwalten von dienstbezogenen Rollen im Allgemeinen finden Sie unter AWS Dienste AWS, die mit IAM funktionieren. Suchen Sie in der Tabelle nach einem Service mit einem `Yes` in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

AWS verwaltete Richtlinien für Amazon CloudWatch Internet Monitor

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird AWS. AWS Verwaltete Richtlinien sind so konzipiert, dass sie Berechtigungen für viele gängige Anwendungsfälle bereitstellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie allen AWS Kunden zur

Verfügung stehen. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie kundenverwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinie: CloudWatchInternetMonitorServiceRolePolicy

Diese Richtlinie ist der dienstbezogenen Rolle zugeordnet, die so benannt AWSServiceRoleForInternetMonitorist, dass Internet Monitor auf Ressourcen in Ihrem Konto zugreifen kann, z. B. Amazon Virtual Private Cloud Cloud-Ressourcen oder Network Load Balancers, sodass Sie diese auswählen können, wenn Sie einen Monitor erstellen. Weitere Informationen finden Sie unter Servicebezogene Rolle für Amazon CloudWatch Internet Monitor.

IAM-Berechtigungen für Amazon CloudWatch Internet Monitor

Um auf Aktionen für die Arbeit mit Monitoren und Daten in Amazon CloudWatch Internet Monitor zugreifen zu können, müssen Benutzer über die richtigen Berechtigungen verfügen.

Weitere Informationen zur Sicherheit bei Amazon CloudWatch finden Sie unterIdentitäts- und Zugriffsmanagement für Amazon CloudWatch.

Berechtigungen für den schreibgeschützten Zugriff in Amazon Internet Monitor CloudWatch

Um auf schreibgeschützte Aktionen für die Arbeit mit Monitoren und Daten in Amazon CloudWatch Internet Monitor zugreifen zu können, müssen Benutzer als Benutzer oder Rolle mit den folgenden Berechtigungen angemeldet sein:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
```

```
                "internetmonitor:Get*",
                "internetmonitor:List*",
                "internetmonitor:StartQuery",
                "internetmonitor:StopQuery",
                "logs:DescribeLogGroups",
                "logs:GetQueryResults",
                "logs:StartQuery",
                "logs:StopQuery"
                ],
            "Resource": "*"
        }
    ]
}
```

Berechtigungen für den vollen Zugriff in Amazon CloudWatch Internet Monitor

Um einen Monitor in Amazon CloudWatch Internet Monitor zu erstellen und vollen Zugriff auf Aktionen für die Arbeit mit Monitoren und Daten in Internet Monitor zu haben, müssen Benutzer mit einem Benutzer oder einer Rolle angemeldet sein, die über die folgenden Berechtigungen verfügt:

- Berechtigungen zum Erstellen einer serviceverknüpften Rolle, die Internet Monitor zugeordnet ist. Weitere Informationen finden Sie unter Servicebezogene Rolle für Amazon CloudWatch Internet Monitor.
- Berechtigungen für Aktionen, die vollen Zugriff auf die Arbeit mit Monitoren und Daten in Internet Monitor ermöglichen.

> **Note**
>
> Wenn Sie eine identitätsbasierte Berechtigungsrichtlinie erstellen, die restriktiver ist, haben Benutzer mit dieser Richtlinie möglicherweise keinen vollständigen Zugriff zum Erstellen von bzw. Arbeiten mit Monitoren und Daten in Internet Monitor.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "internetmonitor:*"
```

```
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
internetmonitor.amazonaws.com/AWSServiceRoleForInternetMonitor",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "internetmonitor.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
internetmonitor.amazonaws.com/AWSServiceRoleForInternetMonitor"
        },
        {
            "Action": [
                "ec2:DescribeVpcs",
                "elasticloadbalancing:DescribeLoadBalancers",
                "workspaces:DescribeWorkspaceDirectories",
                "cloudfront:GetDistribution"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Servicebezogene Rolle für Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor verwendet eine AWS Identity and Access Management (IAM) - Serviceverknüpfte Rolle. Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Internet Monitor verknüpft ist. Die dienstbezogene Rolle ist von Internet Monitor vordefiniert und umfasst alle Berechtigungen, die der Service benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen.

Internet Monitor definiert die Berechtigungen dieser serviceverknüpften Rolle. Sofern keine andere Konfiguration festgelegt wurde, kann nur Internet Monitor die Rolle übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können die Rolle nur nach dem Löschen der zugehörigen Ressourcen löschen. Dies schützt Ihre Internet-Monitor-Ressourcen, da Sie nicht versehentlich die Berechtigungen für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Serviceverknüpfte Rollenberechtigungen für Internet Monitor

Internet Monitor verwendet die angegebene dienstverknüpfte Rolle. AWSServiceRoleForInternetMonitor Diese Rolle ermöglicht Internet Monitor den Zugriff auf Ressourcen in Ihrem Konto, wie Amazon Virtual Private Cloud Cloud-Ressourcen, CloudFront Amazon-Distributionen, WorkSpaces Amazon-Verzeichnisse und Network Load Balancers, sodass Sie diese auswählen können, wenn Sie einen Monitor erstellen.

Diese dienstbezogene Rolle verwendet die verwaltete Richtlinie. `CloudWatchInternetMonitorServiceRolePolicy`

Die AWSServiceRoleForInternetMonitordienstbezogene Rolle vertraut darauf, dass der folgende Dienst die Rolle übernimmt:

- `internetmonitor.amazonaws.com`

Die Berechtigungen für diese Richtlinie finden Sie CloudWatchInternetMonitorServiceRolePolicyin der Referenz für AWS verwaltete Richtlinien.

Erstellen einer serviceverknüpften Rolle für Internet Monitor

Sie müssen die serviceverknüpfte Rolle für Internet Monitor nicht manuell erstellen. Wenn Sie zum ersten Mal einen Monitor erstellen, erstellt Internet Monitor ihn AWSServiceRoleForInternetMonitorfür Sie.

Weitere Informationen finden Sie unter Erstellen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Bearbeiten einer serviceverknüpften Rolle für Internet Monitor

Nachdem Internet Monitor eine serviceverknüpfte Rolle in Ihrem Konto erstellt hat, können Sie den Namen der Rolle nicht mehr ändern, da verschiedene Entitäten auf die Rolle verweisen könnten. Sie können die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für Internet Monitor

Wenn Sie ein Feature oder einen Service nicht mehr benötigen, die bzw. der eine serviceverknüpfte Rolle erfordert, sollten Sie die Rolle löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für die serviceverknüpften Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Nachdem Sie Ihre Ressourcen von Ihren Monitoren in Internet Monitor entfernt und anschließend die Monitore gelöscht haben, können Sie die dienstbezogene Rolle AWSServiceRoleForInternetMonitorlöschen.

> **Note**
>
> Wenn der Internet-Monitor-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies der Fall ist, warten Sie einige Minuten und versuchen Sie es erneut.

So löschen Sie die serviceverknüpfte Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die AWSServiceRoleForInternetMonitordienstverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Updates der serviceverknüpften Rolle von Internet Monitor

Aktualisierungen der AWS verwalteten Richtlinie für die dienstbezogene Rolle Internet Monitor finden Sie unter CloudWatch Updates für AWS verwaltete Richtlinien. AWSServiceRoleForInternetMonitor Abonnieren Sie den RSS-Feed auf der Seite CloudWatch Dokumentverlauf CloudWatch, um automatische Benachrichtigungen über verwaltete Richtlinienänderungen in zu erhalten.

Kontingente in Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor hat die folgenden Kontingente.

Ressource	Standardkontingent
Monitor pro Region	50
Ressourcen pro Monitor	50
Tage, an denen gelöste Internet-Monitor-Zustandsereignisse beibehalten werden	400

Amazon CloudWatch Network Monitor verwenden

Amazon CloudWatch Network Monitor bietet Einblick in die Leistung des Netzwerks, das Ihre AWS gehosteten Anwendungen mit Ihren lokalen Zielen verbindet, und ermöglicht es Ihnen, innerhalb von Minuten die Ursache von Netzwerkleistungseinbußen zu identifizieren. Network Monitor wird vollständig von AWS verwaltet. Sie müssen daher keine zusätzlichen Agenten installieren, um Ihre Netzwerkleistung zu überwachen. Sie können den Paketverlust und die Latenz Ihrer hybriden Netzwerkverbindungen schnell visualisieren, Warnmeldungen und Schwellenwerte festlegen und dann Maßnahmen ergreifen, um das Netzwerkerlebnis Ihrer Endbenutzer zu verbessern.

Network Monitor richtet sich an Netzwerkbetreiber und Anwendungsentwickler, die in Echtzeit Einblicke in die Netzwerkleistung erhalten möchten.

Schlüsselfeatures

- Verwenden Sie Network Monitor, um Ihre sich verändernde hybride Netzwerkumgebung mit kontinuierlichen Messdaten zu Paketverlust und Latenz in Echtzeit zu vergleichen.
- Wenn Sie eine Verbindung herstellen AWS Direct Connect, diagnostiziert Network Monitor schnell eine Netzwerkverschlechterung, indem er den AWS Network Health Indicator in Ihr CloudWatch Konto schreibt. Diese Metrik liefert ein probabilistisches Ergebnis, anhand dessen bestimmt werden kann, ob die Netzwerkverschlechterung innerhalb von AWS lag.
- Network Monitor bietet eine reibungslose Überwachung mit einem vollständig verwalteten Agenten als Ansatz, was bedeutet, dass Sie weder auf VPCs noch On-Premises-Agenten installieren müssen. Sie müssen lediglich ein VPC-Subnetz und eine On-Premises-IP-Adresse angeben, um loszulegen.

- Network Monitor veröffentlicht Metriken unter CloudWatch Metrics. Sie können Dashboards erstellen, um Ihre Metriken anzuzeigen und um umsetzbare Schwellenwerte und Alarme für die für Ihre Anwendung spezifischen Metriken zu erstellen.

Weitere Details finden Sie unter the section called “Funktionsweise von Network Monitor”.

Terminologie und Komponenten von Network Monitor

- Monitor – Ein Monitor zeigt die Ressourcen, für die Sie Netzwerkleistungs- und Verfügbarkeitsmessungen anzeigen und Zustandsereignis-Warnungen erhalten möchten. Wenn Sie einen Monitor für eine Anwendung erstellen, fügen Sie eine AWS gehostete Ressource als Netzwerkquelle hinzu. Network Monitor erstellt dann eine Liste aller möglichen Tests zwischen den AWS gehosteten Ressourcen und Ihren Ziel-IP-Adressen.
- Sonden — Eine Sonde ist der Datenverkehr, der von der AWS gehosteten Ressource an Ihre lokale Ziel-IP-Adresse gesendet wird. Network Monitor-Metriken werden für jede Sonde, die in einem Monitor konfiguriert ist, in Ihr CloudWatch Konto geschrieben.
- AWS Netzwerkquelle — Dies ist die ursprüngliche AWS Quelle eines Netzwerkmonitor-Sondes, bei dem es sich um ein Subnetz in einer Ihrer VPCs handelt.
- Ziel – Dies ist das Ziel in Ihrem On-Premises-Netzwerk für die AWS -Netzwerkquelle. Das Ziel ist eine Kombination aus Ihren On-Premises-IP-Adressen, Netzwerkprotokollen, Ports und der Größe der Netzwerkpakete. Es werden sowohl IPv4- als auch IPv6-Adressen unterstützt.

Einschränkungen und Anforderungen von Network Monitor

- Network Monitor unterstützt maximal vier Ziel-IP-Adressen und bis zu 24 Sonden pro Monitor.
- Sie können bis zu 100 Monitore pro Region pro Konto haben.
- Monitor-Subnetze müssen demselben Konto gehören wie der Monitor.
- Network Monitor bietet kein automatisches Netzwerk-Failover im Falle eines AWS Netzwerkproblems.
- Für jede Sonde, die Sie erstellen, wird eine Gebühr berechnet. Weitere Details finden Sie unter the section called “Preisgestaltung”.

So funktioniert Amazon CloudWatch Network Monitor

Network Monitor erleichtert die Überwachung, indem es eine vollständig verwaltete Lösung ohne Agenten bietet. Wenn Sie in Ihrer AWS gehosteten Ressource einen Monitor erstellen, AWS erstellt und verwaltet er die gesamte Infrastruktur im Hintergrund, um Messungen der Round-Trip-Zeit und des Paketverlusts durchzuführen. Dadurch können Sie Ihre Überwachung schnell skalieren, ohne Agenten in Ihrer AWS Infrastruktur installieren oder deinstallieren zu müssen.

Network Monitor konzentriert sich bei der Überwachung auf die Routen, die von Datenströmen aus Ihren AWS gehosteten Ressourcen genommen werden, anstatt alle Datenflüsse von Ihren AWS-Region Ressourcen umfassend zu überwachen. Wenn sich Ihre Workloads auf mehrere Availability Zones (AZs) verteilen, kann Network Monitor die Routen von jedem Ihrer privaten Subnetze aus überwachen.

Network Monitor veröffentlicht Metriken zu den Rundlaufzeiten und Paketverlusten in Ihrem Amazon CloudWatch -Konto auf der Grundlage des Aggregationsintervalls, das Sie bei der Erstellung eines Monitors festgelegt haben. Sie können auch individuelle Schwellenwerte für Latenz und Paketverlust für jeden verwendeten CloudWatch Monitor festlegen. Sie können beispielsweise einen Alarm einrichten, um Sie zu benachrichtigen, wenn Ihr durchschnittlicher Paketverlust einen statischen Schwellenwert von 0,1 % für einen Workload übersteigt, der für Paketverluste anfällig ist. Sie können die CloudWatch Anomalieerkennung auch verwenden, um bei Messwerten für Paketverlust oder Latenz, die außerhalb der gewünschten Bereiche liegen, einen Alarm auszulösen.

Verfügbarkeits- und Leistungsmessungen

Network Monitor sendet regelmäßig aktive Tests von Ihrer AWS Ressource an Ihre lokalen Ziele. Beim Erstellen eines Monitors geben Sie Folgendes an:

- Das Aggregationsintervall. Die Zeit in Sekunden, in der die CloudWatch Messergebnisse empfangen werden. Dies erfolgt entweder alle 30 oder 60 Sekunden. Der Aggregationszeitraum, den Sie für den Monitor wählen, gilt für alle Sonden in diesem Monitor.
- Das Sonden-Protokoll. Jede Sonde, die einem Monitor hinzugefügt wird, muss entweder das Internet Control Message Protocol (ICMP)- oder das Transmission Control Protocol (TCP)-Protokoll verwenden. Weitere Details finden Sie unter the section called “Kommunikationsprotokolle”.
- Die Paketgröße. Die Größe jedes Pakets in Byte, das zwischen Ihrer AWS gehosteten Ressource und Ihrem Ziel auf einer einzigen Sonde übertragen wird. Jede Sonde in einem Monitor kann ihre eigene Paketgröße haben.

Für Metriken,

- Die in Millisekunden gemessene Metrik zur Zeit für Hin- und Rückfahrt misst und zeichnet eine Leistungsmessung auf und zeichnet die Zeit auf, die benötigt wird, bis die Sonde an die Ziel-IP-Adresse übertragen und die zugehörige Antwort empfangen wird.
- Die Metrik zum Paketverlust misst den Prozentsatz der insgesamt gesendeten Pakete und zeichnet die Anzahl der übertragenen Sonden auf, die keine zugehörige Antwort erhalten haben. Dies bedeutet, dass diese Pakete tatsächlich entlang des Netzwerkpfads verloren gegangen sind.

Unterstützte Kommunikationsprotokolle

ICMP-basierte Sonden leiten ICMP-Echoanfragen von Ihren AWS gehosteten Ressourcen an die Zieladresse weiter und erwarten eine ICMP-Echoantwort von der Zieladresse zurück. Network Monitor verwendet die Informationen der ICMP-Echoanforderung und der ICMP-Echoantwortnachrichten, um die Zeit für Hin- und Rückfahrt und die Paketverlust-Metriken zu berechnen.

TCP-basierte Sonden übertragen TCP-SYN-Pakete von Ihren AWS gehosteten Ressourcen zur Zieladresse und zum Zielport und erwarten ein TCP-SYN+ACK- oder RST-Paket von der Zieladresse und dem Zielport zurück. Network Monitor verwendet die Informationen der TCP-SYN- und TCP-SYN +ACK- bzw. RST-Nachrichten, um die Metriken zur Zeit für Hin- und Rückfahrt und zum Paketverlust zu berechnen. Darüber hinaus wechselt Network Monitor in regelmäßigen Abständen die TCP-Quell-Ports, um die Netzwerkabdeckung zu erhöhen, was wiederum die Wahrscheinlichkeit erhöhen kann, dass Paketverluste erkannt werden.

AWS Indikator für den Netzwerkstatus

Network Monitor veröffentlicht eine Network Health Indicator (NHI)-Metrik, die Informationen zur Netzwerkleistung und Verfügbarkeit für Ziele, die über AWS Direct Connect verbunden sind, liefert. Die Metrik ist ein statistisches Maß für den Zustand des AWS kontrollierten Netzwerkpfads von der AWS gehosteten Ressource, auf der der Monitor bereitgestellt wird, bis zum Direct Connect-Standort.

Network Monitor verwendet die Erkennung von Anomalien, um Verfügbarkeitseinbußen oder Leistungseinbußen entlang Ihrer Netzwerkpfade zu berechnen.

> **Note**
>
> Jedes Mal, wenn Sie einen neuen Monitor erstellen, einen Test hinzufügen oder einen Test erneut aktivieren, wird der NHI für diesen Monitor um einige Stunden verzögert, sodass Daten für die Erkennung von Anomalien AWS gesammelt werden können.

Um die NHI-Zustandsmetrik bereitzustellen, wendet Network Monitor eine statistische Korrelation zwischen den AWS -Beispieldatensätzen sowie auf die Metriken für Paketverlust und Latenz bei Hin- und Rückfahrt für den Datenverkehr an, der Ihren Netzwerkpfad simuliert. Die Metrik kann eine von zwei Variablen sein: 1 oder 0. Ein Wert von 1 gibt an, dass Network Monitor eine Netzwerkverschlechterung innerhalb des AWS kontrollierten Netzwerkpfads beobachtet hat. Ein Wert von 0 gibt an, dass Network Monitor keine Netzwerkverschlechterung innerhalb des Pfads beobachtet hat. Auf diese Weise können Sie Netzwerkprobleme schneller beheben. Sie können Warnmeldungen für die NHI-Metrik einrichten, um über aktuelle Probleme auf Ihren Netzwerkpfaden informiert zu werden.

Support für IPv4- und IPv6-Adressen

Network Monitor bietet Verfügbarkeits- und Leistungsmetriken für IPv4- oder IPv6-Netzwerke und kann entweder IPv4- oder IPv6-Adressen von Dual-Stack-VPCs überwachen. Network Monitor erlaubt nicht, dass sowohl IPv4- als auch IPv6-Ziele innerhalb desselben Monitors konfiguriert werden. Sie können jedoch separate Ziele für reine IPv4- und reine IPv6-Ziele erstellen.

Verfügbarkeit in Regionen

Network Monitor ist derzeit in den folgenden Versionen verfügbar AWS-Regionen:

Region	
Asien-Pazifik (Hongkong)	ap-east-1
Asien-Pazifik (Mumbai)	ap-south-1

Region	
Asien-Pazifik (Seoul)	ap-northeast-2
Asien-Pazifik (Singapur)	ap-southeast-1
Asien-Pazifik (Sydney)	ap-southeast-2
Asien-Pazifik (Tokio)	ap-northeast-1
Kanada West (Calgary)	ca-west-1
Europa (Frankfurt)	eu-central-1
Europa (Irland)	eu-west-1
Europa (London)	eu-west-2
Europa (Paris)	eu-west-3
Europa (Stockholm)	eu-north-1

Region	
Naher Osten (Bahrain)	me-south-1
Südamerika (São Paulo)	sa-east-1
USA Ost (Nord-Virginia)	us-east-1
USA Ost (Ohio)	us-east-2
USA West (Nordkalifornien)	us-west-1
USA West (Oregon)	us-west-2

Erstellen eines Network Monitors

In den folgenden Schritten wird beschrieben, wie Sie einen Monitor erstellen und anschließend die erforderlichen Sonden hinzufügen. Für eine Sonde wählen Sie das Quellsubnetz und bis zu vier Ziel-IP-Adressen für maximal 24 Sonden pro Monitor. Sie können ein Modellpaket entweder über die Amazon CloudWatch -Konsole oder über Befehlszeile oder die API erstellen.

Themen

- Einen Netzwerk-Monitor mithilfe der Konsole erstellen
- Einen Netzwerk-Monitor über die Befehlszeile oder API erstellen

Einen Netzwerk-Monitor mithilfe der Konsole erstellen

In den folgenden Schritten wird das Erstellen eines Monitors mithilfe der Amazon CloudWatch -Konsole beschrieben. Sie wählen Ihre Quellsubnetze aus und fügen dann bis zu vier Ziele hinzu, um bis zu 24 Sonden pro Monitor zu erstellen. Sie können einen Monitor entweder über die Amazon CloudWatch -Konsole oder über die Befehlszeile oder das SDK erstellen.

> **Important**
>
> Diese Schritte sind so konzipiert, dass sie alle auf einmal ausgeführt werden können. Sie können in Bearbeitung befindliche Arbeiten nicht speichern, um sie später fortzusetzen.

Die Überwachungsdetails definieren

Der erste Schritt zur Erstellung eines Monitors besteht darin, die grundlegenden Details zu definieren. Dazu gehört, dem Monitor einen Namen zu geben und den Aggregationszeitraum zu definieren. Sie können optional Tags zum Monitor hinzufügen.

So definieren Sie Monitor-Details

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/ und wählen Sie dann unter Netzwerküberwachung die Option Netzwerkmonitor aus.
2. Klicken Sie auf Create monitor (Überwachung erstellen).
3. Geben Sie für Name des Monitors den Namen ein, den Sie für diesen Monitor verwenden möchten.
4. Wählen Sie für den Aggregationszeitraum aus, an wie oft Sie Messwerte senden möchten. CloudWatch Verfügbare Aggregationszeiträume sind:

   - 30 Sekunden
   - 60 Sekunden

   > **Note**
   >
   > Ein kürzerer Aggregationszeitraum ermöglicht eine schnellere Erkennung von Netzwerkproblemen. Der von Ihnen gewählte Aggregationszeitraum kann sich jedoch auf

Ihre Abrechnungsstruktur auswirken. Weitere Informationen zur Preisgestaltung finden Sie auf der Seite mit den CloudWatch Amazon-Preisen.

5. (Optional) Fügen Sie im Abschnitt Tags Schlüssel- und Wertepaare hinzu, um diese Ressource besser identifizieren zu können, was Ihnen das Suchen oder Filtern nach bestimmten Informationen ermöglicht.

   1. Wählen Sie Neues Tag hinzufügen aus.
   2. Geben Sie einen Schlüsselnamen und den zugehörigen Wert ein.
   3. Wählen Sie Neuen Tag hinzufügen, um diesen neuen Tag hinzuzufügen.

      Sie können mehrere Tags hinzufügen, indem Sie Neuen Tag hinzufügen wählen, oder Sie können ein beliebiges Tag entfernen, indem Sie Entfernen wählen.
   4. Wenn Sie Ihre Tags dem Monitor zuordnen möchten, lassen Sie die Option Tags zu den vom Monitor erstellten Sonden hinzufügen aktiviert. Dadurch werden die Tags zu den Monitor-Sonden hinzugefügt. Dies kann hilfreich sein, wenn Sie die Tag-basierte Authentifizierung oder Messung verwenden.
6. Wählen Sie Weiter zu the section called “Die Quelle und das Ziel wählen”.

Die Quelle und das Ziel wählen

Ein Netzwerkmonitor verwendet eine AWS Quelle für die VPCs und die zugehörigen Subnetze in den Regionen, in denen Ihr Netzwerk betrieben wird. Ein Monitor-Ziel ist die Kombination aus Ihren On-Premises-IP-Adressen, Netzwerkprotokollen, Ports und der Größe der Netzwerkpakete.

Die Kombination aus Quelle und Ziel wird als Sonde bezeichnet. Sie können bis zu vier Sonden pro Subnetz und insgesamt bis zu 24 Sonden pro Monitor haben.

> **Important**
>
> Diese Schritte sind so konzipiert, dass sie alle auf einmal ausgeführt werden können. Sie können in Bearbeitung befindliche Arbeiten nicht speichern, um sie später fortzusetzen.

So wählen Sie eine Quelle und ein Ziel aus

1. Wählen Sie unter AWS-Netzwerkquelle ein oder mehrere Subnetze aus, die in den Monitor aufgenommen werden sollen. Sie können eine einzelne VPC auswählen, die dann alle Subnetze

innerhalb dieser VPC auswählt, oder Sie können bestimmte Subnetze auswählen. Die von Ihnen ausgewählten VPCs und Subnetze dienen als Quelle für den Netzwerk-Monitor.

2. Geben Sie unter Ziel 1 die Ziel-IP-Adresse des On-Premises-Netzwerks ein. Es werden sowohl IPv4- als auch IPv6-Adressen unterstützt.
3. Wählen Sie Erweiterte Einstellungen aus.
4. Wählen Sie für dieses vom Kunden verwaltete Ziel das Netzwerkprotokoll aus. Dies kann folgendes sein:
   - ICMP
   - TCP
5. Wenn das Protokoll TCP ist, geben Sie die folgenden Informationen ein. Andernfalls überspringen Sie diesen Schritt und gehen Sie direkt zum nächsten:
   1. Geben Sie den Port ein, den Ihr Netzwerk für die Verbindung verwendet. Der Port muss eine Zahl zwischen 1 und 65 535 sein.
   2. Geben Sie die Paketgröße ein. Dies ist die Größe jedes Pakets in Byte, das auf der Sonde zwischen Quelle und Ziel gesendet wird. Die Paketgröße muss eine Zahl zwischen 56 und 8 500 sein.
6. Wählen Sie Ziel hinzufügen, um diesem Monitor ein weiteres On-Premises-Ziel hinzuzufügen. Wiederholen Sie diese Schritte für jeden Benutzer, den Sie hinzufügen möchten.
7. Wählen Sie Weiter, wenn Sie fertig sind, um die Sonden zu bestätigen.

Sonden bestätigen

Wenn Sie die Sonden bestätigen, können Sie die Kombination der Netzwerksonden für den Monitor überprüfen. Auf dieser Seite werden alle möglichen Kombinationen der von Ihnen ausgewählten Quellen und Ziele angezeigt. Wenn Sie beispielsweise sechs Quell-Subnetze und vier Ziel-IPs haben, haben Sie insgesamt 24 mögliche Sonden-Kombinationen.

> **Important**
>
> - Diese Schritte sind so konzipiert, dass sie alle auf einmal ausgeführt werden können. Sie können in Bearbeitung befindliche Arbeiten nicht speichern, um sie später fortzusetzen.
> - Auf der Seite Sonden bestätigen wird nicht angegeben, ob eine Sonde gültig ist. Wir empfehlen Ihnen daher, diese Seite gründlich zu überprüfen und alle ungültigen Sonden zu

löschen. Wenn Sie ungültige Sonden nicht entfernen, werden Ihnen diese möglicherweise in Rechnung gestellt.

So bestätigen Sie Überwachungs-Sonden

1. Voraussetzung: the section called “Die Quelle und das Ziel wählen”.
2. Überprüfen Sie auf der Seite Sonden bestätigen die Liste der Quell- und Zielkombinationen.
3. Wählen Sie eine oder mehrere Sonden aus, die Sie vom Monitor entfernen möchten, und wählen Sie dann Entfernen.

   > **Note**
   >
   > Sie werden nicht aufgefordert, den Löschvorgang zu bestätigen. Nachdem eine Sonde gelöscht wurde, muss sie erneut eingerichtet werden. Im Bereich Netzwerk-Monitor auf der Seite Network Monitor können Sie eine Sonde wieder zu einem Monitor hinzufügen. Weitere Informationen finden Sie unter the section called “Einem Monitor eine Sonde hinzufügen”.

4. Wählen Sie Weiter, um die Monitor-Details zu überprüfen, bevor Sie ihn erstellen.

Überprüfen und erstellen

Der letzte Schritt bei der Erstellung eines Monitors und der Sonden besteht darin, die Details sowohl des Monitors als auch der Sonden zu überprüfen. Sie können an dieser Stelle jede der Informationen ändern. Wenn Sie die Überprüfung abgeschlossen und den Monitor erstellt haben und die Erfassung der Metriken beginnt, werden Ihnen alle Sonden in Rechnung gestellt.

> **Important**
>
> - Dieser Schritt ist so konzipiert, dass er bei der Erstellung eines Monitors und einer Sonde auf einmal ausgeführt werden kann. Sie können in Bearbeitung befindliche Arbeiten nicht speichern, um sie später fortzusetzen.
> - Wenn Sie einen Abschnitt bearbeiten möchten, müssen Sie die Erstellung des Monitors ab dem Zeitpunkt, an dem Sie ihn bearbeiten, schrittweise durchführen. Sie müssen

jedoch keine nachfolgenden Schritte erneut erstellen. Auf diesen Seiten werden die zuvor ausgefüllten Informationen beibehalten.

So überprüfen und erstellen Sie einen Monitor

1. Wählen Sie auf der Seite Sonden überprüfen und erstellen für jeden Abschnitt, in dem Sie Änderungen vornehmen möchten, die Option Bearbeiten.
2. Nehmen Sie in diesem Abschnitt sämtliche Änderungen vor.
3. Wählen Sie Weiter aus.
4. Führen Sie eine der folgenden Aktionen aus:
   - Nehmen Sie die gewünschten Änderungen auf zusätzlichen Monitor-Seiten vor und wählen Sie Weiter, bis Sie wieder auf der Seite Überprüfen und erstellen sind.
   - Wenn keine anderen Seiten geändert werden müssen, wählen Sie Weiter, bis Sie wieder auf der Seite Überprüfen und erstellen sind.
5. Klicken Sie auf Create monitor (Überwachung erstellen).

   Auf der Network-Monitor-Seite wird der aktuelle Status der Monitor-Erstellung im Bereich Netzwerk-Monitore angezeigt. Bei der Erstellung des Monitors lautet der Status Ausstehend. Wenn der Status auf Aktiv wechselt, können Sie auf das Monitor-Dashboard zugreifen, um die Messwerte einzusehen CloudWatch .

   Weitere Informationen zum Arbeiten mit dem Monitor-Dashboard finden Sie unter the section called “Dashboards von Network Monitor”.

> **Note**
>
> Möglicherweise dauert es ein paar Minuten, bis der neu hinzugefügte Netzwerk-Monitor mit der Erfassung der Netzwerk-Metriken beginnt.

Einen Netzwerk-Monitor über die Befehlszeile oder API erstellen

Verwenden Sie die Befehlszeile oder API, um einen Netzwerk-Monitor anzuzeigen und zu erstellen.

So erstellen Sie einen Netzwerk-Monitor über die Befehlszeile oder API

1. Erstellen Sie mit create-monitor einen Netzwerk-Monitor.
2. Erstellen Sie mit create-probe eine Netzwerk-Sonde.

Arbeiten mit Monitoren und Sonden von Network Monitor

Sie können jede der folgenden Aufgaben mit Ihren Monitoren und Sonden entweder über die Amazon CloudWatch -Konsole, die Befehlszeile oder die API ausführen.

Themen:

Einen Monitor bearbeiten

Sie können alle Informationen für einen Netzwerk-Monitor bearbeiten, z. B. ihn umbenennen, einen neuen Aggregationszeitraum festlegen oder Tags hinzufügen oder entfernen. Durch das Ändern der Informationen eines Monitors werden keine der zugehörigen Sonden geändert. Sie können einen Monitor entweder über die Amazon CloudWatch -Konsole oder über die Befehlszeile oder die API bearbeiten.

Einen Monitor mithilfe der Konsole bearbeiten

Verwenden Sie die CloudWatch Konsole, um einen Monitor zu bearbeiten.

So bearbeiten Sie einen Monitor mithilfe der Konsole

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/ und wählen Sie dann unter Netzwerküberwachung die Option Netzwerkmonitor aus.
2. Wählen Sie unter Netzwerk-Monitore den Monitor aus, den Sie bearbeiten möchten.

3. Wählen Sie auf der Monitor-Dashboard-Seite die Option Bearbeiten.
4. Geben Sie als Monitornamen den neuen Namen für den Monitor ein.
5. Wählen Sie für den Aggregationszeitraum aus, an wie oft Sie Messwerte senden möchten. CloudWatch Gültige Zeiträume sind:

   - 30 Sekunden
   - 60 Sekunden

   > **Note**
   >
   > Ein kürzerer Aggregationszeitraum ermöglicht eine schnellere Erkennung von Netzwerkproblemen. Der von Ihnen gewählte Aggregationszeitraum kann sich jedoch auf Ihre Abrechnungsstruktur auswirken. Weitere Informationen zur Preisgestaltung finden Sie auf der Seite mit den CloudWatch Amazon-Preisen.

6. (Optional) Fügen Sie im Abschnitt Tags Schlüssel- und Wertepaare hinzu, um diese Ressource besser identifizieren zu können, was Ihnen das Suchen oder Filtern nach bestimmten Informationen ermöglicht. Sie können auch einfach den Wert eines beliebigen aktuellen Schlüssels ändern.

   1. Wählen Sie Neues Tag hinzufügen aus.
   2. Geben Sie einen Schlüsselnamen und den zugehörigen Wert ein.
   3. Wählen Sie Neuen Tag hinzufügen, um diesen neuen Tag hinzuzufügen.

      Sie können mehrere Tags hinzufügen, indem Sie Neuen Tag hinzufügen wählen, oder Sie können ein beliebiges Tag entfernen, indem Sie Entfernen wählen.
   4. Wenn Sie Ihre Tags dem Monitor zuordnen möchten, lassen Sie die Option Tags zu den vom Monitor erstellten Sonden hinzufügen aktiviert. Dadurch werden die Tags zu den Monitor-Sonden hinzugefügt. Dies kann hilfreich sein, wenn Sie die Tag-basierte Authentifizierung oder Messung verwenden.
7. Wählen Sie Änderungen speichern aus.

Einen Monitor über die CLI oder API bearbeiten

Verwenden Sie die Befehlszeile oder die API, um einen Monitor anzuzeigen und zu bearbeiten.

So bearbeiten Sie einen Monitor über die Befehlszeile oder die API

1. Verwenden Sie list-monitors, um eine Liste Ihrer Monitore abzurufen, falls Sie den Namen des Monitors nicht kennen. Notieren Sie den Namen des Monitors, den Sie bearbeiten möchten.
2. Verwenden Sie edit-monitor und verwenden Sie dabei den Namen des Monitors aus dem vorherigen Schritt.

Einen Monitor löschen

Bevor Sie einen Monitor löschen können, müssen Sie alle diesem Monitor zugeordneten Sonden deaktivieren oder löschen, unabhängig vom Status des Monitors. Nachdem ein Monitor deaktiviert oder gelöscht wurde, werden Ihnen für diese Monitor-Sonden keine Gebühren mehr berechnet. Sie können einen gelöschten Monitor nicht wiederherstellen. Sie können einen Monitor entweder über die Amazon CloudWatch Konsole oder über die Befehlszeile/API löschen.

Obwohl ein Test gelöscht oder deaktiviert werden kann, werden die Messwerte CloudWatch dennoch 15 Tage lang gespeichert.

Einen Monitor mithilfe der Konsole löschen

Verwenden Sie die CloudWatch Konsole, um einen Monitor zu löschen.

So löschen Sie einen Monitor mithilfe der Konsole

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/ und wählen Sie dann unter Netzwerküberwachung die Option Netzwerkmonitor aus.
2. Wählen Sie im Abschnitt Netzwerk-Monitore den Monitor aus, den Sie löschen möchten.
3. Wählen Sie Aktionen und anschließend Löschen aus.
4. Wenn Sie aktive Sonden haben, werden Sie aufgefordert, diese zu deaktivieren. Wählen Sie Sonden deaktivieren.

> **Note**
>
> Sie können diese Aktion nicht abbrechen oder rückgängig machen, nachdem Sie Sonden deaktivieren ausgewählt haben. Deaktivierte Sonden werden jedoch nicht aus dem Monitor entfernt. Sie können sie später wieder aktivieren. Siehe the section called “Eine Sonde aktivieren oder deaktivieren”.

5. Geben Sie in das Bestätigungsfeld **`confirm`** ein und wählen Sie dann Löschen.

Einen Monitor über die Befehlszeile oder die API löschen

Löschen Sie einen Monitor über die Befehlszeile oder die API.

So löschen Sie einen Netzwerk-Monitor über die Befehlszeile oder API

1. Sie benötigen den Namen des Monitors, den Sie löschen möchten. Falls Sie den Namen nicht kennen, verwenden Sie list-monitors, um eine Liste Ihrer Monitore abzurufen. Notieren Sie den Namen des Monitors, den Sie löschen möchten.
2. Überprüfen Sie, ob dieser Monitor Sonden enthält. Verwenden Sie get-monitor mit dem Monitor-Namen aus dem vorherigen Schritt. Dadurch wird eine Liste aller Sonden zurückgegeben, die diesem Monitor zugeordnet sind.
3. Wenn der Monitor Sonden enthält, müssen Sie diese Sonden zuerst entweder auf inaktiv setzen oder sie löschen.
   - Um eine Sonde auf inaktiv zu setzen, verwenden Sie update-probe und setzen Sie den Status auf `INACTIVE`.
   - Verwenden Sie delete-probe, um einen Test zu löschen.
4. Sobald die Sonden entweder auf `INACTIVE` gesetzt oder gelöscht sind, verwenden Sie delete-monitor, um den Monitor zu löschen. Inaktive Sonden werden nicht gelöscht.

Eine Sonde aktivieren oder deaktivieren

Sie können eine Monitor-Sonde nach Bedarf aktivieren oder deaktivieren. Möglicherweise möchten Sie eine Sonde deaktivieren, wenn Sie diese derzeit nicht verwenden, sie aber vielleicht in der Zukunft erneut verwenden möchten. Wenn Sie eine Sonde deaktivieren, müssen Sie später keine Zeit damit verbringen, sie erneut einzurichten. Deaktivierte Sonden werden Ihnen nicht in Rechnung gestellt.

Sie können den Status eines Monitors entweder über die Amazon CloudWatch Konsole oder über die Befehlszeile oder API ändern.

Mithilfe der Konsole eine Sonde auf aktiv oder inaktiv setzen

Verwenden Sie die CloudWatch Konsole, um einen Test auf aktiv oder inaktiv zu setzen.

So setzen Sie eine Sonde mithilfe der Konsole auf aktiv oder inaktiv

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/ und wählen Sie dann unter Netzwerküberwachung die Option Netzwerkmonitor aus.
2. Wählen Sie die Registerkarte Monitor-Details.
3. Wählen Sie im Abschnitt Sonden die Sonde aus, die Sie aktivieren oder deaktivieren möchten.
4. Wählen Sie Aktionen und dann entweder Aktivieren oder Deaktivieren.

> **Note**
>
> Wenn Sie eine deaktivierte Sonde reaktivieren, werden Ihnen Gebühren für diese Sonde verrechnet.

Eine Sonde mithilfe der Befehlszeile oder der API auf aktiv oder inaktiv setzen

Setzen Sie eine Sonde mithilfe der Befehlszeile oder der API auf aktiv oder inaktiv. Sie können diesen Befehl nur für eine einzelne Sonde verwenden.

So setzen Sie eine Sonde mithilfe der Befehlszeile oder der API auf aktiv oder inaktiv

1. Verwenden Sie list-monitors, um eine Liste Ihrer Monitore abzurufen, falls Sie den Namen des Monitors nicht kennen. Notieren Sie sich den Namen des Monitors, dessen Sondenstatus Sie ändern möchten.
2. Verwenden Sie get-monitor mit dem Monitor-Namen aus dem vorherigen Schritt. Dadurch wird eine Liste aller Sonden zurückgegeben, die diesem Monitor zugeordnet sind. Notieren Sie sich die Sonden-ID der Sonden, deren Status Sie ändern möchten.
3. Verwenden Sie update-probe und setzen Sie die Sonde, deren Status Sie ändern möchten, entweder auf `ACTIVE` oder `INACTIVE`.

Einem Monitor eine Sonde hinzufügen

Sie können einem vorhandenen Monitor eine Sonde hinzufügen. Beachten Sie, dass Ihre Abrechnungsstruktur aktualisiert wird, wenn Sie einem Monitor Sonden hinzufügen.

Einem Monitor mithilfe der Konsole eine Sonde hinzufügen

So fügen Sie einem Monitor mithilfe der Konsole eine Sonde hinzu

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/ und wählen Sie dann unter Netzwerküberwachung die Option Netzwerkmonitor aus.
2. Führen Sie im Abschnitt Netzwerk-Monitore einen der folgenden Schritte aus:
   - Wählen Sie den Link Name des Monitors aus, dem eine Sonde hinzugefügt werden soll. Wählen Sie die Registerkarte Monitor-Details und wählen Sie dann im Abschnitt Sonden die Option Sonde hinzufügen.
   - Aktivieren Sie das Monitor-Kontrollkästchen, wählen Sie Aktionen und anschließend Sonde hinzufügen.
3. Führen Sie auf der Seite Sonde hinzufügen die folgenden Schritte aus:
   1. Wählen Sie unter AWS-Netzwerkquelle ein Subnetz aus, das dem Monitor hinzugefügt werden soll.

      > **Note**
      >
      > Sie können jeweils nur eine Sonde und bis zu vier Sonden pro Monitor hinzufügen.

   2. Geben Sie die Ziel-IP-Adresse des On-Premises-Netzwerks ein. Es werden sowohl IPv4- als auch IPv6-Adressen unterstützt.
   3. Wählen Sie Erweiterte Einstellungen aus.
   4. Wählen Sie das Netzwerkprotokoll für das Ziel aus. Dies kann entweder ICMP oder TCP sein.
   5. Wenn das Protokoll TCP ist, geben Sie die folgenden Informationen ein. Andernfalls überspringen Sie diesen Schritt und gehen Sie direkt zum nächsten:
      - Geben Sie den Port ein, den Ihr Netzwerk für die Verbindung verwendet. Der Port muss eine Zahl zwischen 1 und 65 535 sein.
      - Geben Sie die Paketgröße ein. Dabei handelt es sich um die Größe jedes Pakets in Byte, das auf der Sonde zwischen Quelle und Ziel gesendet wird. Die Paketgröße muss eine Zahl zwischen 56 und 8 500 sein.
4. (Optional) Fügen Sie im Abschnitt Tags Schlüssel- und Wertepaare hinzu, um diese Ressource besser identifizieren zu können, was Ihnen das Suchen oder Filtern nach bestimmten Informationen ermöglicht.

1. Wählen Sie Neues Tag hinzufügen aus.
2. Geben Sie einen Schlüsselnamen und den zugehörigen Wert ein.
3. Wählen Sie Neuen Tag hinzufügen, um den neuen Tag hinzuzufügen.

   Sie können mehrere Tags hinzufügen, indem Sie Neuen Tag hinzufügen wählen, oder Sie können ein beliebiges Tag entfernen, indem Sie Entfernen wählen.

5. Wählen Sie Sonde hinzufügen.

   Während die Sonde aktiviert wird, wird der Status als Ausstehend angezeigt. Möglicherweise dauert es ein paar Minuten, bis die Sonde aktiv wird.

Einem Monitor über die Befehlszeile oder die API eine Sonde hinzufügen

Fügen Sie einem Monitor über die Befehlszeile oder API eine Sonde hinzu. Sie können diesen Befehl nur verwenden, um jeweils eine einzelne Sonde hinzuzufügen.

So fügen Sie einem Monitor über die Befehlszeile oder die API eine Sonde hinzu

1. Verwenden Sie list-monitors, um eine Liste Ihrer Monitore abzurufen, falls Sie den Namen des Monitors nicht kennen. Notieren Sie sich den Namen des Monitors, dem Sie eine Sonde hinzufügen möchten.
2. Verwenden Sie create-probe, um dem Monitor eine Sonde hinzuzufügen.

Eine Sonde bearbeiten

Sie können alle Informationen für eine aktuelle Sonde ändern, unabhängig davon, ob diese Sonde aktiviert oder deaktiviert ist. Sie können eine Sonde entweder über die Amazon CloudWatch -Konsole oder über die Befehlszeile oder die API bearbeiten.

Eine Sonde mit der Konsole bearbeiten

So bearbeiten Sie eine Sonde mit der Konsole

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/ und wählen Sie dann unter Netzwerküberwachung die Option Netzwerkmonitor aus.

   Wählen Sie den Link Name, um das Monitor-Dashboard zu öffnen.
2. Wählen Sie die Registerkarte Monitor-Details.

3. Wählen Sie im Abschnitt Sonden den Link für die Sonde aus, die Sie bearbeiten möchten.
4. Wählen Sie auf der Sonden-Dashboard-Seite entweder Bearbeiten oder Aktionen und dann Bearbeiten.
5. Geben Sie auf der Seite Sonde bearbeiten die neue IP-Adresse der Zielsonde ein. Es werden sowohl IPv4- als auch IPv6-Adressen unterstützt.
6. Wählen Sie Erweiterte Einstellungen aus.
7. Wählen Sie das Netzwerkprotokoll aus. Dies kann entweder ICMP oder TCP sein.
8. Wenn das Protokoll TCP ist, geben Sie die folgenden Informationen ein. Andernfalls überspringen Sie diesen Schritt und gehen Sie direkt zum nächsten:
   - Geben Sie den Port ein, den Ihr Netzwerk für die Verbindung verwendet. Der Port muss eine Zahl zwischen 1 und 65 535 sein.
   - Geben Sie die Paketgröße ein. Dabei handelt es sich um die Größe jedes Pakets in Byte, das auf der Sonde zwischen Quelle und Ziel gesendet wird. Die Paketgröße muss eine Zahl zwischen 56 und 8 500 sein.
9. (Optional) Fügen Sie im Abschnitt Tags Schlüssel- und Wertepaare hinzu, um diese Ressource besser identifizieren zu können, was Ihnen das Suchen oder Filtern nach bestimmten Informationen ermöglicht.
   1. Wählen Sie Neues Tag hinzufügen aus.
   2. Geben Sie einen Schlüsselnamen und den zugehörigen Wert ein.
   3. Wählen Sie Neuen Tag hinzufügen, um den neuen Tag hinzuzufügen.

      Sie können mehrere Tags hinzufügen, indem Sie Neuen Tag hinzufügen wählen, oder Sie können ein beliebiges Tag entfernen, indem Sie Entfernen wählen.
10. Wählen Sie Änderungen speichern aus.

Eine Sonde über die Befehlszeile oder die API bearbeiten

Verwenden Sie die Befehlszeile, um eine Monitor-Sonde zu bearbeiten. Sie können diesen Befehl nur für eine einzelne Sonde verwenden.

So bearbeiten Sie eine Sonde über die Befehlszeile oder die API

1. Verwenden Sie list-monitors, um eine Liste Ihrer Monitore abzurufen, falls Sie den Namen des Monitors nicht kennen. Notieren Sie sich den Namen des Monitors, dessen Sondenstatus Sie ändern möchten.
2. Verwenden Sie get-monitor mit dem Monitor-Namen aus dem vorherigen Schritt. Dadurch wird eine Liste aller Sonden zurückgegeben, die diesem Monitor zugeordnet sind. Notieren Sie die Sonden-ID einer Sonde, die Sie bearbeiten möchten.
3. Verwenden Sie update-probe, um die Informationen der Sonde zu ändern.

Eine Sonde löschen

Sie können eine Sonde löschen, anstatt sie zu deaktivieren, wenn Sie wissen, dass Sie sie in Zukunft nicht mehr benötigen werden. Sie können eine gelöschte Sonde nicht wiederherstellen und müssten sie stattdessen neu erstellen. Die Abrechnung für diese Sonde wird beendet, wenn die Sonde gelöscht wird. Sie können eine Sonde entweder über die Amazon CloudWatch -Konsole oder über die Befehlszeile oder die API löschen.

Eine Sonde mithilfe der Konsole löschen

So löschen Sie eine Sonde mithilfe der Konsole

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/, und wählen Sie dann unter Netzwerküberwachung die Option Netzwerkmonitor aus.
2. Wählen Sie im Bereich Netzwerk-Monitore den Link Name, um das Monitor-Dashboard zu öffnen.
3. Wählen Sie die Registerkarte Monitor-Details.
4. Wählen Sie das Monitor-Kontrollkästchen, dann Aktionen und anschließend Löschen.
5. Wählen Sie im Dialogfeld Sonde löschen die Option Löschen, um zu bestätigen, dass Sie die Sonde löschen möchten.
6. Wählen Sie Löschen, um zu bestätigen, dass Sie die Sonde löschen möchten.

   Der Status der Sonde wird im Abschnitt Sonden als Wird gelöscht angezeigt. Nach dem Löschen wird die Sonde aus dem Abschnitt Sonden entfernt.

Eine Sonde über die Befehlszeile oder die API löschen

Löschen einer Sonde über die Befehlszeile oder die API. Sie können diesen Befehl nur für eine einzelne Sonde verwenden.

So setzen Sie eine Sonde mithilfe der Befehlszeile oder der API auf aktiv oder inaktiv

1. Verwenden Sie list-monitors, um eine Liste Ihrer Monitore abzurufen, falls Sie den Namen des Monitors nicht kennen. Den Namen des Monitors notieren, dessen Sonde Sie löschen möchten
2. Verwenden Sie get-monitor mit dem Monitor-Namen aus dem vorherigen Schritt. Dadurch wird eine Liste aller Sonden zurückgegeben, die diesem Monitor zugeordnet sind. Notieren Sie die Sonden-ID der Sonde, die Sie löschen möchten.
3. Verwenden Sie delete-probe.

Ressourcen über die Befehlszeile oder die API mit Tags versehen bzw. diese aufheben

Sie können die Befehlszeile oder die CLI verwenden, um Ressourcen-Tags hinzuzufügen oder zu aktualisieren.

So aktualisieren Sie Network-Monitor-Tags über die Befehlszeile oder API

- Um Ressourcen-Tags aufzulisten, verwenden Sie list-tags-for-resources.
- Verwenden Sie tag-resource, um eine Ressource zu markieren.
- Verwenden Sie untag-resource, um das Tag einer Ressource aufzuheben.

Dashboards von Network Monitor

Sie können das Amazon CloudWatch Network Monitor-Dashboard verwenden, um den AWS Netzwerkstatus zu überprüfen und die Round-Trip-Zeit und den Paketverlust zu untersuchen. Sie können diese Metriken sowohl für Monitore als auch für einzelne Sonden anzeigen.

Dashboards von Network Monitor

- Monitor-Dashboard
- Sonden-Dashboard

Sondenalarme

Sie können CloudWatch Amazon-Alarme auf der Grundlage von Amazon CloudWatch Network Monitor-Metriken erstellen, genau wie Sie es für andere CloudWatch Amazon-Metriken tun können. Jeder Alarm, den Sie erstellen, wird in der Spalte Status der Sonde im Abschnitt Monitor-Details des Network Monitor-Dashboards angezeigt, wenn der Alarm ausgelöst wird. Der Status lautet entweder „OK“ oder „Bei Alarm“. Wenn für eine Sonde kein Status angezeigt wird, wurde für diese Sonde kein Alarm ausgelöst.

Sie können z. B. einen Alarm auf der Grundlage der Network-Monitor-Metrik `PacketLoss` erstellen und ihn so konfigurieren, dass er eine Benachrichtigung sendet, wenn die Metrik über einem von Ihnen gewählten Wert liegt. Sie konfigurieren Alarme für Network Monitor-Metriken nach den gleichen Richtlinien wie für andere CloudWatch Messwerte.

Die folgenden Messwerte sind unter `AWS/NetworkMonitor` Beim Erstellen eines CloudWatch Alarms für Network Monitor verfügbar.

- HealthIndicator
- PacketLoss
- RTT (Zeit für Rundumlauf)

Die Schritte zum Erstellen eines Network-Monitor-Alarms finden Sie unter the section called “Erstellen eines Alarms basierend auf einem statischen Schwellenwert”.

Festlegen eines Zeitrahmens für Metriken

Für Metriken und Ereignisse auf beiden Dashboards wird eine Standardzeit von zwei Stunden verwendet, die ausgehend von der aktuellen Uhrzeit berechnet wird. Sie können die Standardeinstellung ändern, sodass eine der folgenden Voreinstellungen verwendet wird:

- 1h – eine Stunde
- 2h – zwei Stunden
- 1d – ein Tag
- 1w – eine Woche

Sie können auch einen benutzerdefinierten Zeitrahmen festlegen. Wählen Sie Benutzerdefiniert, wählen Sie eine Absolute oder Relative Zeit und legen Sie dann den Zeitrahmen auf eine Zeit Ihrer

Wahl fest. Die relative Zeit unterstützt CloudWatch standardmäßig nur einen Zeitraum von 15 Tagen ab dem heutigen Datum.

Darüber hinaus können Sie die in den Diagrammen angezeigte Uhrzeit entweder auf der Grundlage der UTC-Zeitzone oder einer lokalen Zeitzone auswählen.

Monitor-Dashboard

Sie können das Amazon CloudWatch Network Monitor-Dashboard verwenden, um den AWS Netzwerkstatus zu überprüfen und die Round-Trip-Zeit und den Paketverlust zu untersuchen. Network Monitor verfügt über Dashboards sowohl für Monitore als auch für Sonden.

So greifen Sie auf ein Monitor-Dashboard zu

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/ und wählen Sie dann unter Netzwerküberwachung die Option Netzwerkmonitor.
2. Wählen Sie im Bereich Netzwerk-Monitore den Link Name, um das Monitor-Dashboard zu öffnen.

Übersicht

Die Übersichtsseite zeigt die folgenden Informationen für Ihren Monitor an:

- AWS Netzwerkintegrität — AWS Netzwerkintegrität zeigt nur den Gesamtzustand des AWS Netzwerks an. Der Status lautet entweder Gesund oder Degradiert. Der Status Fehlerfrei bedeutet, dass der Netzwerkmonitor keine Probleme mit dem AWS Netzwerk festgestellt hat. Der Status Heruntergestuft bedeutet, dass Network Monitor ein Problem mit dem AWS Netzwerk festgestellt hat. Die Statusleiste in diesem Abschnitt zeigt den Status des Netzwerks über einen Standardzeitraum von einer Stunde an. Bewegen Sie den Mauszeiger über einen beliebigen Punkt der Statusleiste, um weitere Details anzuzeigen.
- Zusammenfassung des Sondenverkehrs — Zeigt den aktuellen Status des Datenverkehrs zwischen den AWS Quellsubnetzen im Monitor und den Ziel-IP-Adressen an. In der Zusammenfassung des Sonden-Datenverkehrs wird Folgendes angezeigt:
  - Sonden im Alarmzustand – Diese Zahl gibt an, wie viele Ihrer Sonden sich in einem degradierten Zustand befinden. Ein Alarm wird ausgelöst, wenn eine Metrik ausgelöst wird, die Sie als Alarm eingerichtet haben. Informationen zu metrischen Alarmen von Network Monitor finden Sie unterthe section called “Sondenalarme”.

- Paketverlust – Die Anzahl der Pakete, die vom Quellsubnetz zur Ziel-IP-Adresse verloren gegangen sind. Dies wird als Prozentsatz der insgesamt gesendeten Pakete dargestellt.
- Zeit für Hin- und Rückfahrt – Die Zeit in Millisekunden, die ein Paket aus dem Quellsubnetz benötigt, um die Ziel-IP-Adresse zu erreichen und dann wieder zurückzukommen.

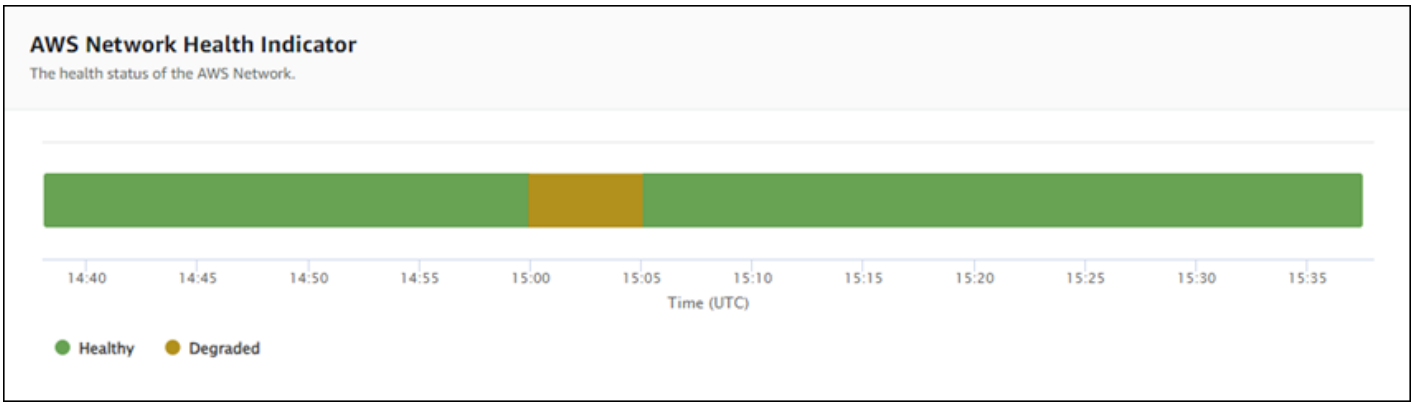
Die Daten werden in einem interaktiven Diagramm dargestellt, sodass Sie die Details sehen können.

Standardmäßig werden Daten für einen Zeitraum von zwei Stunden angezeigt, der anhand des aktuellen Datums und der aktuellen Uhrzeit berechnet wird. Sie können den Bereich jedoch entsprechend Ihren Anforderungen ändern. Weitere Informationen finden Sie unter the section called “Festlegen eines Zeitrahmens für Metriken”.

Verfolgen von Metriken

Das Network-Monitor-Dashboard zeigt eine grafische Darstellung Ihrer Monitore und Sonden. Folgende Diagramme sind verfügbar:

- AWS Netzwerkintegritätsindikator — Dies stellt den Zustand des AWS Netzwerks über einen bestimmten Zeitraum dar. Der Status lautet entweder Gesund oder Beeinträchtigt. Im folgenden Beispiel sehen Sie, dass sich das AWS Netzwerk von 15:00 Uhr UTC bis 15:05 UTC in einem heruntergekommenen Zustand befand. Nach 15:05 Uhr kehrte das Netzwerk in einen fehlerfreien Zustand zurück. Sie können den Mauszeiger über einen beliebigen Abschnitt des Diagramms bewegen, um weitere Details anzuzeigen.



> **Note**
>
> Der Network Health Indicator gibt nicht den Zustand der Sonde an, sondern nur den Zustand des AWS Netzwerks.

- Paketverlust – Dieses Diagramm zeigt eine eindeutige Linie, die den Prozentsatz des Paketverlusts für jede Sonde im Monitor anzeigt. In der Legende unten auf der Seite werden alle Sonden auf dem Monitor angezeigt, wobei die einzelnen Sonden farblich gekennzeichnet sind, um ihre Eindeutigkeit zu gewährleisten. Wenn Sie in diesem Diagramm mit der Maus auf eine Sonde zeigen, werden das Quellsubnetz, die Ziel-IP und der Prozentsatz des Paketverlusts angezeigt. Im folgenden Beispiel wurde ein Paketverlust-Alarm für eine Sonde von einem Subnetz zur IP-Adresse 127.0.0.1 eingerichtet. Der Alarm wurde ausgelöst, als der Schwellenwert für den Paketverlust für die Sonde überschritten wurde. Wenn Sie den Mauszeiger über das Diagramm bewegen, werden die Quelle und das Ziel der Sonde angezeigt. Außerdem wird angezeigt, dass bei dieser Sonde am 21. November um 02:41:30 Uhr ein Paketverlust von 30,97 % zu verzeichnen war.

- Zeit für Rundumlauf – In diesem Diagramm wird für jede Sonde eine Linie angezeigt, die die Zeit für den Rundumlauf für jede Sonde anzeigt. In der Legende unten auf der Seite werden alle Sonden auf dem Monitor angezeigt, wobei die einzelnen Sonden farblich gekennzeichnet sind, um ihre Eindeutigkeit zu gewährleisten. Wenn Sie in diesem Diagramm mit der Maus auf eine Sonde zeigen, werden das Quellsubnetz, die Ziel-IP-Adresse und die Zeit für den Rundumlauf angezeigt. Das folgende Beispiel zeigt, dass am Dienstag, den 21. November, um 21:45:30 Uhr die Zeit für Hin- und Rückfahrt für eine Sonde von einem Subnetz zur IP-Adresse 127.0.0.1 0,075 Sekunden betrug.

Monitor-Details

Auf der Seite mit den Monitor-Details werden die Details zu Ihrem Monitor einschließlich der Sonden angezeigt. Auf dieser Seite können Sie Tags verwalten oder eine Sonde hinzufügen. Diese Seite ist in die folgenden drei Abschnitte gegliedert:

- Monitor-Details – Diese Seite enthält Details zu Ihrem Monitor. Die Informationen in diesem Abschnitt können nicht bearbeitet werden. Sie können jedoch den Link Rollenname wählen, um Details der serviceverknüpften Network-Monitor-Rolle anzuzeigen.
- Sonden – In diesem Abschnitt wird eine Liste aller dem Monitor zugeordneten Sonden angezeigt. Wählen Sie einen VPC- oder Subnetz-ID-Link, um die VPC- oder Subnetz-Details in der Amazon-VPC-Konsole zu öffnen. Sie können eine Sonde auch ändern und sie aktivieren oder deaktivieren. Weitere Informationen finden Sie unter the section called “Arbeiten mit Monitoren und Sonden”.

  Im Abschnitt Tests werden Informationen zu jeder Sonde angezeigt, die für diesen Monitor eingerichtet wurden, einschließlich der Sonde-ID, der VPC-ID, der Subnetz-ID, der IP-Adresse, des Protokolls und ob der Sondenstatus Aktiv oder Inaktiv ist. Wenn Sie einen Alarm für eine Sonde eingerichtet haben, wird der aktuelle Status dieses Alarms angezeigt. OK bedeutet, dass keine Messwerte vorhanden sind. Ereignisse haben Alarme ausgelöst. Bei Alarm bedeutet, dass eine Metrik, die Sie in eingerichtet haben, einen Alarm CloudWatch ausgelöst hat. Wenn für eine Sonde kein Status angezeigt wird, wurde kein CloudWatch Alarm eingerichtet. Informationen zu den Typen von Network Monitor-Sondenalarmen, die Sie erstellen können, finden Sie unterthe section called “Sondenalarme”.
- Tags – Zeigt die aktuellen Tags für einen Monitor an. Sie können Tags hinzufügen oder entfernen, indem Sie Tags verwalten wählen. Dadurch wird die Seite Sonde bearbeiten geöffnet. Weitere

Informationen zum Bearbeiten von Tags finden Sie unter the section called “Einen Monitor bearbeiten”.

Sonden-Dashboard

Sie können das Amazon CloudWatch Network Monitor-Dashboard verwenden, um den AWS Netzwerkstatus sowie Informationen zu bestimmten Round-Trip-Zeiten und Paketverlusten für bestimmte Tests anzuzeigen. Es gibt zwei Sonden-Dashboards: Übersicht und Sonden-Details.

Sie können CloudWatch Alarme erstellen, um Metrik-Schwellenwerte für Paketverlust und Round-Trip-Zeit festzulegen. Wenn ein Schwellenwert für eine Metrik erreicht wird, werden Sie durch einen CloudWatch Alarm benachrichtigt. Weitere Informationen zum Erstellen von Sonden-Alarmen finden Sie unter the section called “Sondenalarme”.

So greifen Sie auf ein Sonden-Dashboard zu

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/ und wählen Sie dann unter Netzwerküberwachung die Option Netzwerkmonitor aus.
2. Wählen Sie im Bereich Netzwerk-Monitore den Link Name, um das Monitor-Dashboard zu öffnen.
3. Wählen Sie den ID-Link, um das Dashboard für diese Sonde anzuzeigen.

Übersicht

Die Übersichtsseite zeigt die folgenden Informationen für Ihre Sonde an:

- AWS Details zum Netzwerkintegritätsindikator — Dieser gibt nur den Gesamtzustand des AWS Netzwerks an. Der Status lautet entweder Gesund oder Degradiert. Der Status Heruntergestuft weist auf ein AWS Netzwerkproblem hin und gibt nicht an, ob ein Problem mit Ihrer Sonde vorliegt.
- Paketverlust – Die Anzahl der Pakete, die für diese Sonde vom Quellsubnetz zur Ziel-IP-Adresse verloren gegangen sind.
- Zeit für Hin- und Rückfahrt – Die Zeit in Millisekunden, die ein Paket aus dem Quellsubnetz benötigt, um die Ziel-IP-Adresse zu erreichen und dann wieder zurückzukommen.

Sonden-Details

Auf der Seite mit den Sonden-Details werden die Details zu einer Sonde angezeigt. Auf dieser Seite können Sie die Sonde bearbeiten. Weitere Informationen finden Sie unter the section called “Arbeiten mit Monitoren und Sonden”.

- Sonden-Details – Diese Seite enthält allgemeine Informationen zur Sonde. Die Informationen in diesem Abschnitt können nicht bearbeitet werden.
- Quelle und Ziel der Sonde – In diesem Abschnitt werden Details zur Sonde angezeigt. Wählen Sie einen VPC- oder Subnetz-ID-Link, um die VPC- oder Subnetz-Details in der Amazon-VPC-Konsole zu öffnen. Sie können eine Sonde auch ändern und sie aktivieren oder deaktivieren.
- Tags – Zeigt die aktuellen Tags für einen Monitor an. Sie können Tags hinzufügen oder entfernen, indem Sie Tags verwalten wählen. Dadurch wird die Seite Sonde bearbeiten geöffnet. Weitere Informationen zum Bearbeiten von Tags finden Sie unter the section called “Eine Sonde bearbeiten”.

Network-Monitor-Kontingente

Im Folgenden sind die Network-Monitor-Kontingente aufgeführt:

Kontingent	Standard	Anpassbar
Maximale Anzahl von Monitoren pro Konto pro AWS-Region	100	Ja
Maximale Anzahl von Sonden pro Monitor	24	Ja
Maximale Anzahl von Sonden pro Subnetz und Monitor	4	Ja

Datensicherheit und Datenschutz in Network Monitor

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, auf der AWS Dienste in der ausgeführt AWS Cloud werden. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Weitere Informationen zu den Compliance-Programmen, die für Amazon CloudWatch Network Monitor gelten, finden Sie unter AWS Services im Umfang nach Compliance-Programm AWS .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Service, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Verwendung von CloudWatch Network Monitor anwenden können. In den folgenden Themen erfahren Sie, wie Sie CloudWatch Network Monitor so konfigurieren, dass Sie Ihre Sicherheits- und Compliance-Ziele erreichen. Sie erfahren auch, wie Sie andere AWS Dienste verwenden können, die Sie bei der Überwachung und Sicherung Ihrer CloudWatch Network Monitor-Ressourcen unterstützen.

Themen

Datenschutz in Amazon CloudWatch Network Monitor

Das AWS Modell der mit gilt für den Datenschutz in Amazon CloudWatch Network Monitor. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle Systeme laufen AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS -Modell der geteilten Verantwortung und in der DSGVO im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM)

einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie SSL/TLS, um mit Ressourcen zu kommunizieren. AWS Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein. AWS CloudTrail
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-2.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit CloudWatch Network Monitor oder anderen Geräten arbeiten und die Konsole, die API oder SDKs AWS-Services verwenden. AWS CLI AWS Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Infrastruktursicherheit in Amazon CloudWatch Network Monitor

Als verwalteter Service ist Amazon CloudWatch Network Monitor durch die AWS globalen Netzwerksicherheitsverfahren geschützt, die im Whitepaper Amazon Web Services: Sicherheitsprozesse im Überblick beschrieben sind.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das CloudWatch Netzwerk auf Network Monitor zuzugreifen. Kunden müssen Transport Layer Security (TLS) 1.0 oder neuer unterstützen. Wir empfehlen TLS 1.2 oder neuer. Clients müssen außerdem Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) unterstützen. Die meisten modernen Systemen wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Identitäts- und Zugriffsmanagement für Amazon CloudWatch Network Monitor

AWS Identity and Access Management (IAM) ist ein AWS Dienst, der einem Administrator hilft, den Zugriff auf AWS Ressourcen sicher zu kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um CloudWatch Network Monitor-Ressourcen zu verwenden. IAM ist ein AWS Dienst, den Sie ohne zusätzliche Kosten nutzen können. Sie können IAM-Funktionen verwenden, um anderen Benutzern, Services und Anwendungen die uneingeschränkte oder eingeschränkte Nutzung Ihrer AWS -Ressourcen zu erlauben, ohne Ihre Sicherheitsanmeldeinformationen zu teilen.

IAM-Benutzer sind standardmäßig nicht berechtigt, AWS -Ressourcen zu erstellen, anzuzeigen oder zu ändern. Damit ein IAM-Benutzer auf Ressourcen wie ein globales Netzwerk zugreifen und Aufgaben ausführen kann, müssen Sie Folgendes tun:

- Eine IAM-Richtlinie erstellen, die dem Benutzer die Berechtigung zur Nutzung der spezifischen Ressourcen und API-Aktionen erteilt, die er benötigt
- Die Richtlinie dem IAM-Benutzer oder der Gruppe zuweisen, zu der der IAM-Benutzer gehört

Wenn Sie einem Benutzer oder einer Benutzergruppe eine Richtlinie zuweisen, wird den Benutzern die Ausführung der angegebenen Aufgaben für die angegebenen Ressourcen gestattet oder verweigert.

Bedingungsschlüssel

Das `Condition`-Element (auch Bedingungs-Block genannt) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Bedingungselement ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungs-Operatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingungsoperatoren im Benutzerhandbuch für AWS Identity and Access Management.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist.

Sie können Tags an CloudWatch Network Monitor-Ressourcen anhängen oder Tags in einer Anfrage an Cloud WAN übergeben. Um den Zugriff auf Basis von Tags zu steuern, geben Sie Tag-Informationen im Bedingungselement einer Richtlinie mithilfe der Bedingungsschlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` oder `aws:TagKeys` an. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im Benutzerhandbuch für AWS Identity and Access Management.

Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im AWS Identity and Access Management-Benutzerhandbuch.

Kern-Netzwerkressourcen markieren

Ein Tag ist eine Metadaten-Bezeichnung, die Sie oder einer AWS Ressource AWS zuweisen. Jedes Tag besteht aus einem Schlüssel und einem Wert. Für Tags, die Sie zuweisen, definieren Sie einen Schlüssel und einen Wert. So können Sie beispielsweise den Schlüssel als `purpose` und den Wert für eine Ressource als `test` definieren. Tags sind für folgende Aktivitäten nützlich:

- Identifizieren und organisieren Sie Ihre AWS Ressourcen. Viele AWS Dienste unterstützen Tagging, sodass Sie Ressourcen aus verschiedenen Diensten dasselbe Tag zuweisen können, um anzuzeigen, dass die Ressourcen miteinander verknüpft sind.
- Kontrollieren Sie den Zugriff auf Ihre AWS Ressourcen. Weitere Informationen finden Sie unter Steuern des Zugriffs auf AWS Ressourcen mithilfe von Tags im AWS Identify and Access Management-Benutzerhandbuch.

So funktioniert Amazon CloudWatch Network Monitor mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf CloudWatch Network Monitor zu verwalten, sollten Sie sich darüber informieren, welche IAM-Funktionen für Network Monitor verfügbar sind. CloudWatch

IAM-Funktionen, die Sie mit Amazon CloudWatch Network Monitor verwenden können

IAM-Feature	CloudWatch Unterstützung für Network Monitor
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Bedingungsschlüssel für die Richtlinie	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Teilweise
Temporäre Anmeldeinformationen	Ja
Hauptberechtigungen	Ja
Servicerollen	Nein
Serviceverknüpfte Rollen	Ja

Einen allgemeinen Überblick darüber, wie CloudWatch Network Monitor und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren.

Identitätsbasierte Richtlinien für Amazon Network Monitor CloudWatch

Unterstützt Richtlinien auf Identitätsbasis.	Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für Network Monitor CloudWatch

Beispiele für identitätsbasierte Richtlinien von CloudWatch Network Monitor finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amazon CloudWatch

Ressourcenbasierte Richtlinien in Network Monitor CloudWatch

Unterstützt ressourcenbasierte Richtlinien	Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM-Administrator des vertrauenswürdigen Kontos auch der Prinzipalentität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter Wie sich IAM-Rollen von ressourcenbasierten Richtlinien unterscheiden im IAM-Benutzerhandbuch.

Richtlinienaktionen für CloudWatch Network Monitor

Unterstützt Richtlinienaktionen	Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der CloudWatch Network Monitor-Aktionen finden Sie unter Von Amazon CloudWatch Network Monitor definierte Aktionen in der Service Authorization Reference.

Richtlinienaktionen in CloudWatch Network Monitor verwenden vor der Aktion das folgende Präfix:

```
networkmonitor
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
      "networkmonitor:action1",
      "networkmonitor:action2"
         ]
```

Beispiele für identitätsbasierte Richtlinien von CloudWatch Network Monitor finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amazon CloudWatch

Richtlinienressourcen für Network Monitor CloudWatch

Unterstützt Richtlinienressourcen	Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein – `Resource`oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Eine Liste der CloudWatch Network Monitor-Ressourcentypen und ihrer ARNs finden Sie unter Von Amazon CloudWatch Network Monitor definierte Ressourcen in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter Von Amazon CloudWatch Network Monitor definierte Aktionen.

Schlüssel zur Richtlinienbedingung für CloudWatch Network Monitor

Unterstützt servicespezifische Richtlinienbedingungsschlüssel	Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen `AND`-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die

Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

Eine Liste der CloudWatch Network Monitor-Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für Amazon CloudWatch Network Monitor in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von Amazon CloudWatch Network Monitor definierte Aktionen.

ACLs im CloudWatch Netzwerkmonitor

Unterstützt ACLs	Nein

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit Netzwerkmonitor CloudWatch

Unterstützt ABAC (Tags in Richtlinien)	Teilweise

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In AWS werden diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Was ist ABAC? im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

Verwenden temporärer Anmeldeinformationen mit CloudWatch Network Monitor

Unterstützt temporäre Anmeldeinformationen	Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services funktionieren AWS-Services , finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter Wechseln zu einer Rolle (Konsole) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldeinformationen in IAM.

Serviceübergreifende Prinzipalberechtigungen für CloudWatch Network Monitor

Unterstützt Forward Access Sessions (FAS)	Ja

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

Dienstrollen für CloudWatch Network Monitor

Unterstützt Servicerollen	Nein

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

> **Warning**
>
> Durch das Ändern der Berechtigungen für eine Servicerolle kann die Funktionalität von CloudWatch Network Monitor beeinträchtigt werden. Bearbeiten Sie Servicerollen nur, wenn CloudWatch Network Monitor Sie dazu anleitet.

Verwenden einer dienstbezogenen Rolle für CloudWatch Network Monitor

Unterstützt serviceverknüpfte Rollen	Ja

Eine dienstverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene

Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie in der Tabelle nach einem Service mit einem `Yes` in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Beispiele für identitätsbasierte Richtlinien für Network Monitor CloudWatch

Standardmäßig sind Benutzer und Rollen nicht berechtigt, CloudWatch Netzwerkmonitor-Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben mithilfe der AWS API AWS Management Console, AWS Command Line Interface (AWS CLI) oder ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Einzelheiten zu den von CloudWatch Network Monitor definierten Aktionen und Ressourcentypen, einschließlich des Formats der ARNs für jeden Ressourcentyp, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon CloudWatch Network Monitor in der Service Authorization Reference.

Themen

- Bewährte Methoden für Richtlinien
- Verwenden der Network Monitor-Konsole CloudWatch
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer
- Problembehandlung bei Identität und Zugriff auf CloudWatch Network Monitor

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand CloudWatch Network Monitor-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um Ihren Benutzern und Workloads zunächst Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter AWS -verwaltete Richtlinien oder AWS -verwaltete Richtlinien für Auftrags-Funktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung zum IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Konfigurieren eines MFA-geschützten API-Zugriffs im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der Network Monitor-Konsole CloudWatch

Um auf die Amazon CloudWatch Network Monitor-Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den CloudWatch Network Monitor-Ressourcen in Ihrem aufzulisten und einzusehen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen die CloudWatch Network Monitor-Konsole weiterhin verwenden können, fügen Sie den Entitäten auch den CloudWatch Netzwerkmonitor *ConsoleAccess* oder die *ReadOnly* AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zu einem Benutzer im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API AWS CLI oder AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
```

```
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Problembehandlung bei Identität und Zugriff auf CloudWatch Network Monitor

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit CloudWatch Network Monitor und IAM auftreten können.

Themen

- Ich bin nicht autorisiert, eine Aktion in CloudWatch Network Monitor durchzuführen
- Ich bin nicht berechtigt, iam durchzuführen: PassRole
- Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine CloudWatch Network Monitor-Ressourcen ermöglichen

Ich bin nicht autorisiert, eine Aktion in CloudWatch Network Monitor durchzuführen

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven *`my-example-widget`*-Ressource anzuzeigen, jedoch nicht über `networkmonitor:`*`GetWidget`*-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 networkmonitor:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `networkmonitor:GetWidget`-Aktion auf die *`my-example-widget`*-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam durchzuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht autorisiert sind, die `iam:PassRole` Aktion auszuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an CloudWatch Network Monitor übergeben können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion im CloudWatch Netzwerkmonitor auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine CloudWatch Network Monitor-Ressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien

oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob CloudWatch Network Monitor diese Funktionen unterstützt, finden Sie unterSo CloudWatch arbeitet Amazon mit IAM.
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinien für CloudWatch Network Monitor

Um Benutzern, Gruppen und Rollen Berechtigungen hinzuzufügen, ist es einfacher, AWS verwaltete Richtlinien zu verwenden, als Richtlinien selbst zu schreiben. Es erfordert Zeit und Fachwissen, um von Kunden verwaltete IAM-Richtlinien zu erstellen, die Ihrem Team nur die benötigten Berechtigungen bieten. Um schnell loszulegen, können Sie unsere AWS verwalteten Richtlinien verwenden. Diese Richtlinien decken allgemeine Anwendungsfälle ab und sind in Ihrem AWS Konto verfügbar. Weitere Informationen zu AWS verwalteten Richtlinien finden Sie unter AWS Verwaltete Richtlinien im IAM-Benutzerhandbuch.

AWS Dienste verwalten und aktualisieren AWS verwaltete Richtlinien. Sie können die Berechtigungen in AWS verwalteten Richtlinien nicht ändern. Services fügen einer von AWS verwalteten Richtlinien gelegentlich zusätzliche Berechtigungen hinzu, um neue Features zu unterstützen. Diese Art von Update betrifft alle Identitäten (Benutzer, Gruppen und Rollen), an welche die Richtlinie angehängt ist. Services aktualisieren eine von AWS verwaltete Richtlinie am ehesten, ein neues Feature gestartet wird oder neue Vorgänge verfügbar werden. Dienste entfernen keine Berechtigungen aus einer AWS verwalteten Richtlinie, sodass durch Richtlinienaktualisierungen Ihre bestehenden Berechtigungen nicht beeinträchtigt werden.

AWS Unterstützt außerdem verwaltete Richtlinien für Jobfunktionen, die sich über mehrere Dienste erstrecken. Die `ReadOnlyAccess` AWS verwaltete Richtlinie bietet beispielsweise schreibgeschützten Zugriff auf alle AWS Dienste und Ressourcen. Wenn ein Dienst eine neue Funktion startet, werden nur Leseberechtigungen für neue Operationen und Ressourcen AWS hinzugefügt. Eine Liste und Beschreibungen der Richtlinien für Auftragsfunktionen finden Sie in Verwaltete AWS -Richtlinien für Auftragsfunktionen im IAM-Leitfaden.

AWS verwaltete Richtlinie: CloudWatchNetworkMonitorServiceRolePolicy

Die `CloudWatchNetworkMonitorServiceRolePolicy` ist einer dienstbezogenen Rolle zugeordnet, die es dem Dienst ermöglicht, Aktionen in Ihrem Namen durchzuführen und auf Ressourcen zuzugreifen, die mit CloudWatch Network Monitor verknüpft sind. Sie können diese Richtlinie nicht an Ihre IAM-Identitäten anfügen. Weitere Informationen finden Sie unter the section called “Service-verknüpfte Rollen”.

CloudWatch Die Netzwerküberwachung aktualisiert die AWS verwalteten Richtlinien

Hier finden Sie Informationen zu Aktualisierungen der AWS verwalteten Richtlinien für die CloudWatch Netzwerküberwachung, seit dieser Dienst im November 2023 damit begonnen hat, diese Änderungen zu verfolgen.

Änderung	Beschreibung	Datum
CloudWatchNetworkMonitorServiceRolePolicy: Neue Richtlinie.	Dem CloudWatch Netzwerkmonitor wurde eine neue Richtlinie hinzugefügt.	8. November 2023
the section called “`AWSServiceRoleForNetworkMonitor`”. Neue Rolle.	Dem CloudWatch Netzwerkmonitor wurde eine neue Rolle hinzugefügt.	8. November 2023

IAM-Berechtigungen für CloudWatch Network Monitor

Um Amazon CloudWatch Network Monitor verwenden zu können, müssen Benutzer über die richtigen Berechtigungen verfügen.

Weitere Informationen zur Sicherheit bei Amazon CloudWatch finden Sie unterIdentitäts- und Zugriffsmanagement für Amazon CloudWatch.

Für die Anzeige eines Monitors sind Berechtigungen erforderlich

Um einen Monitor für Amazon CloudWatch Network Monitor in der anzeigen zu können AWS Management Console, müssen Sie als Benutzer oder Rolle mit den folgenden Berechtigungen angemeldet sein:

```
{
"Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "networkmonitor:Get*",
                "networkmonitor:List*"
                ],
            "Resource": "*"
        }
    ]
}
```

Erforderliche Berechtigungen für die Erstellung eines Monitors

Um einen Monitor in Amazon CloudWatch Network Monitor zu erstellen, müssen Benutzer über die Berechtigung verfügen, eine serviceverknüpfte Rolle zu erstellen, die mit Network Monitor verknüpft ist. Weitere Informationen zur serviceverknüpften Rolle finden Sie unter Verwenden Sie eine dienstbezogene Rolle für CloudWatch Network Monitor.

Um einen Monitor für Amazon CloudWatch Network Monitor in der zu erstellen AWS Management Console, müssen Sie als Benutzer oder Rolle angemeldet sein, der über die in der folgenden Richtlinie enthaltenen Berechtigungen verfügt.

> **Note**
>
> Wenn Sie eine identitätsbasierte Berechtigungsberechtigungsrichtlinie erstellen, die restriktiver ist, können Benutzer mit dieser Richtlinie keine Überwachung erstellen.

```
{
"Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "networkmonitor:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
networkmonitor.amazonaws.com/AWSServiceRoleForNetworkMonitor",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "networkmonitor.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:GetRole",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
networkmonitor.amazonaws.com/AWSServiceRoleForNetworkMonitor"
        },
        {
            "Action": [
                "ec2:CreateSecurityGroup",
                "ec2:CreateNetworkInterface",
                "ec2:CreateTags"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Verwenden Sie eine dienstbezogene Rolle für CloudWatch Network Monitor

Amazon CloudWatch Network Monitor verwendet die folgende dienstbezogene Rolle für die Berechtigungen, die erforderlich sind, um andere AWS Dienste in Ihrem Namen aufzurufen:

- `AWSServiceRoleForNetworkMonitor`

`AWSServiceRoleForNetworkMonitor`

CloudWatch Network Monitoring verwendet die angegebene dienstbezogene Rolle, um `AWSServiceRoleForNetworkMonitor` CloudWatch Netzwerkmonitore zu aktualisieren und zu verwalten.

Die serviceverknüpfte Rolle `AWSServiceRoleForNetworkMonitor` vertraut darauf, dass der folgende Service die Rolle annimmt:

- `networkmonitor.amazonaws.com`

`CloudWatchNetworkMonitorServiceRolePolicy` ist der serviceverknüpften Rolle angefügt und gewährt dem Service Zugriff auf VPC- und EC2-Ressourcen in Ihrem Konto sowie auf die Verwaltung der erstellten Netzwerk-Monitors.

Berechtigungsgruppen

Die Richtlinie wird in die folgenden Berechtigungsgruppen eingeteilt:

- **`cloudwatch`**- Auf diese Weise kann der Dienstprinzipal Netzwerküberwachungsmetriken für CloudWatch Ressourcen veröffentlichen.
- **`ec2`** – Dies erlaubt es dem Service-Prinzipal, VPCs und Subnetze in Ihrem Konto zu beschreiben, um Monitore und Sonden zu erstellen oder zu aktualisieren. Dies erlaubt es dem Service-Prinzipal auch, Sicherheitsgruppen, Netzwerkschnittstellen und die zugehörigen Berechtigungen zu erstellen, zu ändern und zu löschen, um den Monitor oder die Sonde so zu konfigurieren, dass Überwachungs-Datenverkehr an Ihre Endpunkte gesendet wird.

Weitere Informationen über die Richtlinie finden Sie unter the section called “AWS verwaltete Richtlinien”.

Im Folgenden wird die `CloudWatchNetworkMonitorServiceRolePolicy` dargestellt:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "PublishCw",
   "Effect": "Allow",
   "Action": "cloudwatch:PutMetricData",
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "cloudwatch:namespace": "AWS/NetworkMonitor"
    }
   }
  },
  {
   "Sid": "DescribeAny",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribeNetworkInterfaceAttribute",
    "ec2:DescribeVpcs",
    "ec2:DescribeNetworkInterfacePermissions",
    "ec2:DescribeSubnets",
    "ec2:DescribeSecurityGroups"
   ],
   "Resource": "*"
  },
  {
   "Sid": "DeleteModifyEc2Resources",
   "Effect": "Allow",
   "Action": [
    "ec2:AuthorizeSecurityGroupEgress",
    "ec2:CreateNetworkInterfacePermission",
    "ec2:DeleteNetworkInterfacePermission",
    "ec2:RevokeSecurityGroupEgress",
    "ec2:ModifyNetworkInterfaceAttribute",
    "ec2:DeleteNetworkInterface",
    "ec2:DeleteSecurityGroup"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:network-interface/*",
    "arn:aws:ec2:*:*:security-group/*"
   ],
```

```
    "Condition": {
     "StringEquals": {
      "aws:ResourceTag/ManagedByCloudWatchNetworkMonitor": "true"
     }
    }
   }
  ]
}
```

Erstellen der serviceverknüpften Rolle

`AWSServiceRoleForNetworkMonitor`

Sie müssen die `AWSServiceRoleForNetworkMonitor`-Rolle nicht manuell erstellen.

- CloudWatch Network Monitor erstellt die `AWSServiceRoleForNetworkMonitor` Rolle, wenn Sie Ihren ersten Netzwerkmonitor erstellen. Diese Rolle gilt für alle nachfolgenden Monitore, die Sie erstellen.

Sie müssen über die erforderlichen Berechtigungen verfügen, um eine serviceverknüpfte Rolle in Ihrem Namen zu erstellen. Weitere Informationen finden Sie unter Berechtigungen für serviceverknüpfte Rollen im IAM-Benutzerhandbuch.

Bearbeiten der serviceverknüpften Rolle

Sie können die Beschreibung der `AWSServiceRoleForNetworkMonitor` mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen der serviceverknüpften Rolle

Wenn Sie CloudWatch Network Monitor nicht mehr verwenden müssen, empfehlen wir Ihnen, die `AWSServiceRoleForNetworkMonitor` Rolle zu löschen.

Sie können diese serviceverknüpfte Rollen erst löschen, nachdem Sie den Netzwerk-Monitor gelöscht haben. Informationen zum Löschen des Netzwerk-Monitors finden Sie unter Einen Netzwerk-Monitor löschen.

Sie können die IAM-Konsole, die IAM-CLI oder die IAM-API verwenden, um serviceverknüpfte Rollen zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Nach dem Löschen erstellt `AWSServiceRoleForNetworkMonitor` CloudWatch Network Monitor die Rolle erneut, wenn Sie einen neuen Monitor erstellen.

Unterstützte Regionen für die dienstbezogene CloudWatch Netzwerkmonitor-Rolle

CloudWatch Network Monitor unterstützt die dienstbezogene Rolle überall dort, AWS-Regionen wo der Dienst verfügbar ist. Weitere Informationen finden Sie unter AWS -Endpunkte in der Allgemeine AWS-Referenz.

Löschen der serviceverknüpften Rolle

Wenn Sie CloudWatch Network Monitor nicht mehr verwenden müssen, empfehlen wir Ihnen, die `AWSServiceRoleForNetworkMonitor` Rolle zu löschen.

Sie können diese serviceverknüpfte Rollen erst löschen, nachdem Sie den Netzwerk-Monitor gelöscht haben. Informationen zum Löschen des Netzwerk-Monitors finden Sie unter Einen Netzwerk-Monitor löschen.

Sie können die IAM-Konsole, die IAM-CLI oder die IAM-API verwenden, um serviceverknüpfte Rollen zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Nach dem Löschen erstellt `AWSServiceRoleForNetworkMonitor` CloudWatch Network Monitor die Rolle erneut, wenn Sie einen neuen Monitor erstellen.

Preisgestaltung

Mit Amazon CloudWatch Network Monitor fallen keine Vorabkosten oder langfristigen Verpflichtungen an. Die Preisgestaltung für Network Monitor setzt sich aus den folgenden zwei Komponenten zusammen:

- eine Gebühr pro Stunde pro überwachter Ressource und
- CloudWatch Metriken, Gebühren.

Wenn Sie einen Netzwerk-Monitor erstellen, ordnen Sie ihm Ressourcen zu, die überwacht werden sollen. Für Network Monitor sind dies Subnetze in Ihrer Amazon Virtual Private Cloud (VPC). Mit jeder überwachten Ressource können Sie bis zu vier Sonden von jedem Subnetz in Ihren VPCs zu vier Zielen erstellen. Um Ihre Rechnung unter Kontrolle zu halten, können Sie die Subnetzabdeckung und die On-Premises-IP-Abdeckung anpassen, indem Sie die Anzahl Ihrer überwachten Ressourcen reduzieren.

Weitere Informationen zur Preisgestaltung finden Sie auf der Seite mit den CloudWatch Amazon-Preisen.

Überwachung der Infrastruktur

In den Themen in diesem Abschnitt werden CloudWatch Funktionen erläutert, die Ihnen helfen können, betriebliche Einblicke in Ihre AWS Ressourcen zu erhalten.

Themen

Container Insights

Verwenden Sie CloudWatch Container Insights, um Metriken und Logs aus Ihren containerisierten Anwendungen und Microservices zu sammeln, zu aggregieren und zusammenzufassen. Container Insights sind für die Plattformen von Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS) und Kubernetes auf Amazon EC2 verfügbar. Container Insights unterstützt das Sammeln von Metriken aus Clustern, AWS Fargate die sowohl für Amazon ECS als auch für Amazon EKS bereitgestellt werden.

CloudWatch sammelt automatisch Metriken für viele Ressourcen wie CPU, Arbeitsspeicher, Festplatte und Netzwerk. Container Insights bietet auch Diagnoseinformationen, wie z. B.Fehler beim Container-Neustart, damit Sie Probleme schnell aufdecken und beheben können. Sie können auch CloudWatch Alarme für Metriken einrichten, die Container Insights sammelt.

Container Insights sammelt Daten als Leistungsprotokollereignisse unter Verwendung des eingebetteten Metrikformats. Diese Leistungsprotokollereignisse sind Einträge, die ein strukturiertes JSON-Schema verwenden, das die Aufnahme und Speicherung von Daten mit hoher Kardinalität ermöglicht. Aus diesen Daten werden aggregierte Metriken auf Cluster-, Knoten-, Pod-, Aufgaben- und Serviceebene als CloudWatch Metriken CloudWatch erstellt. Die von Container Insights gesammelten Metriken sind in CloudWatch automatischen Dashboards verfügbar und können auch im Bereich Metriken der Konsole eingesehen werden. CloudWatch Die Metriken sind erst sichtbar, wenn die Container-Aufgaben bereits einige Zeit laufen.

Wenn Sie Container Insights einsetzen, erstellt es automatisch eine Protokollgruppe für die Leistungsprotokoll-Ereignisse. Sie müssen diese Protokollgruppe nicht selbst erstellen.

Um Ihnen bei der Verwaltung Ihrer Container Insights-Kosten zu helfen, erstellt CloudWatch nicht automatisch alle möglichen Metriken aus den Protokolldaten. Sie können jedoch zusätzliche Metriken und zusätzliche Granularitätsebenen anzeigen, indem Sie CloudWatch Logs Insights verwenden, um die rohen Performance-Log-Ereignisse zu analysieren.

In der Originalversion von Container Insights werden erfasste Metriken und aufgenommene Protokolle als benutzerdefinierte Metriken berechnet. Bei Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS werden die Container-Insights-Metriken und -Protokolle pro Beobachtung abgerechnet, anstatt pro gespeicherter Metrik oder aufgenommenem Protokoll. Weitere Informationen zur CloudWatch Preisgestaltung finden Sie unter CloudWatch Amazon-Preise.

In Amazon EKS und Kubernetes verwendet Container Insights eine containerisierte Version des CloudWatch Agenten, um alle laufenden Container in einem Cluster zu ermitteln. Anschließend sammelt es Leistungsdaten auf jeder Ebene des Performance-Stacks.

Container Insights unterstützt die Verschlüsselung mit der AWS KMS key für die gesammelten Protokolle und Metriken. Um diese Verschlüsselung zu aktivieren, müssen Sie die AWS KMS Verschlüsselung für die Protokollgruppe, die Container Insights-Daten empfängt, manuell aktivieren. Dies veranlasst Container Insights, diese Daten mit dem angegebenen KMS-Schlüssel zu verschlüsseln. Es werden nur symmetrische Schlüssel unterstützt. Verwenden Sie keine asymmetrischen KMS-Schlüssel, um Ihre Protokollgruppen zu verschlüsseln.

Weitere Informationen finden Sie unter Verschlüsseln von Protokolldaten in CloudWatch Logs Using AWS KMS.

Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS

Am 6. November 2023 wurde eine neue Version von Container Insights veröffentlicht. Diese Version unterstützt die erweiterte Beobachtbarkeit für Amazon-EKS-Cluster, die auf Amazon EC2 ausgeführt werden, und kann detailliertere Metriken aus diesen Clustern erfassen. Nach der Installation sammelt sie automatisch detaillierte Infrastrukturtelemetrie- und Container-Protokolle für Ihre Amazon-EKS-Cluster. Anschließend können Sie kuratierte, sofort verwendbare Dashboards nutzen, um die Anwendungs- und Infrastrukturtelemetrie genauer zu untersuchen.

Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS erfasst detaillierte Zustands-, Leistungs- und Statusmetriken bis hin zur Container-Ebene sowie Metriken auf Steuerebene. Weitere

Informationen zu den zusätzlichen Metriken und Dimensionen, die erfasst werden können, finden Sie unter Container-Insights-Metriken für Amazon EKS und Kubernetes.

Wenn Sie Container Insights nach dem 6. November 2023 mithilfe des CloudWatch Agenten auf einem Amazon EKS-Cluster auf Amazon EC2 installiert haben, verfügen Sie über Container Insights mit verbesserter Observability für Amazon EKS. Andernfalls können Sie einen Amazon-EKS-Cluster auf diese neue Version aktualisieren, indem Sie den Anweisungen unter Upgrade auf Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS folgen.

Container Insights unterstützt CloudWatch kontenübergreifende Beobachtbarkeit. Sie verwenden ein einziges Monitoring-Konto, um Ihre Anwendungen zu überwachen und Fehler zu beheben, die sich über mehrere AWS Konten innerhalb einer einzigen Region erstrecken. Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.

Container Insights mit verbesserter Observability für Amazon EKS unterstützt auch Windows-Worker-Knoten.

Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS wird auf Fargate nicht unterstützt.

> **Note**
>
> Sie können herausfinden, ob Sie Cluster haben, die auf Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS aktualisiert werden können, indem Sie zur Container-Insights-Konsole navigieren. Wählen Sie dazu im Navigationsbereich der CloudWatch Konsole Insights, Container Insights aus. In der Container-Insights-Konsole informiert Sie ein Banner darüber, ob Sie Amazon-EKS-Cluster haben, die aktualisiert werden können, und enthält Links zur Upgrade-Seite.

Unterstützte Plattformen

Container Insights sind für die Plattformen von Amazon Elastic Container Service, Amazon Elastic Kubernetes Service und Kubernetes auf Amazon-EC2-Instances verfügbar.

- Für Amazon ECS sammelt Container Insights Metriken auf Cluster-, Aufgaben- und Service-Ebene sowohl auf Linux- als auch auf Windows Server-Instances. Es kann Metriken auf Instance-Ebene nur auf Linux-Instances sammeln.

Für Amazon ECS sind Netzwerkmetriken nur für Container im `bridge`- und `awsvpc`-Netzwerkmodus verfügbar. Sie sind nicht für Container im `host`-Netzwerkmodus verfügbar.

- Für Amazon Elastic Kubernetes Service und Kubernetes-Plattformen auf Amazon-EC2-Instances wird Container Insights nur unter Linux-Instances unterstützt.

CloudWatch Agenten-Container-Image

Amazon stellt ein CloudWatch Agenten-Container-Image in Amazon Elastic Container Registry bereit. Weitere Informationen finden Sie unter cloudwatch-agent auf Amazon ECR.

Unterstützte Regionen

Container Insights für Amazon ECS wird in den folgenden Regionen unterstützt:

- USA Ost (Nord-Virginia)
- USA Ost (Ohio)
- USA West (Nordkalifornien)
- USA West (Oregon)
- Africa (Cape Town)
- Asien-Pazifik (Hongkong)
- Asien-Pazifik (Hyderabad)
- Asien-Pazifik (Jakarta)
- Asien-Pazifik (Mumbai)
- Asia Pacific (Osaka)
- Asia Pacific (Seoul)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Tokio)
- Asien-Pazifik (Sydney)
- Kanada West (Calgary)
- Kanada (Zentral)
- Europe (Frankfurt)
- Europa (Irland)

- Europa (London)
- Europa (Milan)
- Europa (Paris)
- Europa (Spain)
- Europa (Stockholm)
- Europa (Zürich)
- Naher Osten (Bahrain)
- Naher Osten (VAE)
- Südamerika (São Paulo)
- AWS GovCloud (USA-Ost)
- AWS GovCloud (US-West)
- China (Peking)
- China (Ningxia)

Unterstützte Regionen für Amazon EKS und Kubernetes

Container Insights für Amazon EKS und Kubernetes wird in den folgenden Regionen unterstützt:

- USA Ost (Nord-Virginia)
- USA Ost (Ohio)
- USA West (Nordkalifornien)
- USA West (Oregon)
- Asien-Pazifik (Hongkong)
- Asien-Pazifik (Mumbai)
- Asia Pacific (Seoul)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)
- Canada (Central)
- China (Peking)
- China (Ningxia)
- Europe (Frankfurt)

- Europa (Irland)
- Europe (London)
- Europe (Paris)
- Europe (Stockholm)
- Middle East (Bahrain)
- Südamerika (São Paulo)
- AWS GovCloud (US-Ost)
- AWS GovCloud (US-West)

Einrichten von Container Insights

Der Container-Insights-Einrichtungsvorgang ist für Amazon ECS und Amazon EKS und für Kubernetes verschieden.

Themen

- Einrichten von Container Insights für Amazon ECS
- Einrichten von Container Insights in Amazon EKS und Kubernetes

Einrichten von Container Insights für Amazon ECS

Sie können eine oder beide der folgenden Optionen verwenden, um Container Insights auf Amazon-ECS-Clustern zu aktivieren:

- Verwenden Sie das AWS Management Console oder das AWS CLI , um mit der Erfassung von Metriken auf Cluster-, Aufgaben- und Service-Ebene zu beginnen.
- Stellen Sie den CloudWatch Agenten als Daemon-Service bereit, um mit der Erfassung von Metriken auf Instance-Ebene auf Clustern zu beginnen, die auf Amazon EC2 EC2-Instances gehostet werden.

Themen

- Einrichten von Container Insights in Amazon ECS für Cluster- und Service-Level-Metriken
- Einrichtung von Container Insights auf Amazon ECS mit AWS Distro für OpenTelemetry
- Bereitstellung des CloudWatch Agenten zur Erfassung von Metriken auf EC2-Instanzebene auf Amazon ECS

- Bereitstellung der AWS Distribution für die Erfassung von Metriken OpenTelemetry auf EC2-Instanzebene auf Amazon ECS-Clustern
- Richten Sie FireLens das Senden von Protokollen an CloudWatch Logs ein

Einrichten von Container Insights in Amazon ECS für Cluster- und Service-Level-Metriken

Sie können Container Insights für neue und vorhandene Amazon-ECS-Cluster aktivieren. Container Insights sammelt Metriken auf Cluster-, Aufgaben- und Service-Ebenen. Sie können Container Insights entweder über die Amazon ECS-Konsole oder die aktivieren AWS CLI.

Wenn Sie Amazon ECS auf einer Amazon-EC2-Instance verwenden und Netzwerk- und Speichermetriken aus Container Insights sammeln möchten, starten Sie diese Instance mit einem AMI, das den Amazon-ECS-Agenten Version 1.29 enthält. Weitere Informationen zum Aktualisieren Ihrer Agenten-Version finden Sie unter Aktualisieren des Amazon-ECS-Container-Agenten.

Sie können das verwenden AWS CLI , um Berechtigungen auf Kontoebene festzulegen, um Container Insights für alle neuen Amazon ECS-Cluster zu aktivieren, die in Ihrem Konto erstellt wurden. Geben Sie dazu den folgenden Befehl ein.

```
aws ecs put-account-setting --name "containerInsights" --value "enabled"
```

> **Note**
>
> Wenn der vom Kunden verwaltete AWS KMS Schlüssel, den Sie für Ihre Amazon ECS Container Insights-Metriken verwenden, noch nicht für die Verwendung konfiguriert ist CloudWatch, müssen Sie die Schlüsselrichtlinie aktualisieren, um verschlüsselte CloudWatch Protokolle in Logs zuzulassen. Sie müssen außerdem Ihren eigenen AWS KMS Schlüssel mit der Protokollgruppe unter verknüpfen`/aws/ecs/containerinsights/`*`ClusterName`*`/performance`. Weitere Informationen finden Sie unter Verschlüsseln von Protokolldaten in CloudWatch Logs using AWS Key Management Service.

Einrichten von Container Insights für vorhandene Amazon-ECS-Cluster

Um Container Insights für einen vorhandenen Amazon-ECS-Cluster zu aktivieren, geben Sie den folgenden Befehl ein. Sie müssen Version 1.16.200 oder höher von ausführen, AWS CLI damit der folgende Befehl funktioniert.

```
aws ecs update-cluster-settings --cluster myCICluster --settings
 name=containerInsights,value=enabled
```

Einrichten von Container Insights für neue Amazon-ECS-Cluster

Es gibt zwei Möglichkeiten, Container Insights für neue Amazon-ECS-Cluster zu aktivieren. Sie können Amazon ECS so konfigurieren, dass alle neuen Cluster standardmäßig für Container Insights aktiviert sind. Andernfalls können Sie einen neuen Cluster aktivieren, wenn Sie ihn erstellen.

Verwenden des AWS Management Console

Sie können Container Insights standardmäßig für alle neuen Cluster oder für einen einzelnen Cluster einstellen, während Sie ihn erstellen.

So stellen Sie Container Insights standardmäßig für alle neuen Cluster ein

1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/ecs/v2.
2. Wählen Sie im Navigationsbereich Account Settings (Kontoeinstellungen).
3. Wählen Sie Aktualisieren.
4. Um CloudWatch Container Insights standardmäßig für Cluster zu verwenden, aktivieren oder deaktivieren Sie unter CloudWatchContainer Insights CloudWatch Container Insights.
5. Wählen Sie Änderungen speichern aus.

Wenn Sie Container Insights nicht mit dem voranstehenden Verfahren standardmäßig für alle neuen Clustern aktiviert haben, gehen Sie wie folgt vor, um einen Cluster zu erstellen, bei dem Container Insights aktiviert ist.

So erstellen Sie einen Cluster mit aktivem Container Insights

1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/ecs/v2.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie auf der Seite Clusters die Option Create cluster (Cluster erstellen) aus.
4. Geben Sie unter Cluster configuration (Cluster-Konfiguration) für Cluster name (Clustername) einen eindeutigen Namen ein.

   Der Name kann bis zu 255 Buchstaben (Groß- und Kleinbuchstaben), Ziffern und Bindestriche enthalten.

5. Um Container Insights zu aktivieren, erweitern Sie Überwachen und aktivieren Sie dann Container Insights verwenden.

Sie können jetzt Aufgabendefinitionen erstellen, Aufgaben ausführen und Services im Cluster starten. Weitere Informationen finden Sie hier:

- Erstellen einer Aufgabendefinition
- Ausführen von Aufgaben
- Erstellen eines Service

Einrichtung von Container Insights auf neuen Amazon ECS-Clustern mit dem AWS CLI

Wenn Container Insights auf allen neuen Cluster standardmäßig aktiviert werden soll, geben Sie den folgenden Befehl ein.

```
aws ecs put-account-setting --name "containerInsights" --value "enabled"
```

Wenn Sie Container Insights nicht mit dem voranstehenden Befehl standardmäßig für alle neuen Cluster aktiviert haben, geben Sie zum Erstellen eines neuen Clusters, bei dem Container Insights aktiviert ist, den folgenden Befehl ein. Sie müssen Version 1.16.200 oder höher der AWS CLI ausführen, damit der folgende Befehl funktioniert.

```
aws ecs create-cluster --cluster-name myCICluster --settings
 "name=containerInsights,value=enabled"
```

Deaktivieren von Container Insights in Amazon-ECS-Clustern

Um Container Insights für einen vorhandenen Amazon-ECS-Cluster zu deaktivieren, geben Sie den folgenden Befehl ein.

```
aws ecs update-cluster-settings --cluster myCICluster --settings
 name=containerInsights,value=disabled
```

Einrichtung von Container Insights auf Amazon ECS mit AWS Distro für OpenTelemetry

Verwenden Sie diesen Abschnitt, wenn Sie AWS Distro for verwenden möchten OpenTelemetry , um CloudWatch Container Insights auf einem Amazon ECS-Cluster einzurichten. Weitere Informationen zu AWS Distro for Open Telemetry finden Sie unter AWS Distro for. OpenTelemetry

Bei diesen Schritten wird davon ausgegangen, dass Sie bereits über einen Cluster verfügen, auf dem Amazon ECS ausgeführt wird. Weitere Informationen zur Verwendung von AWS Distro for Open Telemetry mit Amazon ECS und zur Einrichtung eines Amazon ECS-Clusters für diesen Zweck finden Sie unter Setting up AWS Distro for OpenTelemetry Collector in Amazon Elastic Container Service.

Schritt 1: Erstellen einer Aufgabenrolle

Der erste Schritt besteht darin, eine Aufgabenrolle im Cluster zu erstellen, die der AWS OpenTelemetry Collector verwenden wird.

Um eine Aufgabenrolle für AWS Distro zu erstellen für OpenTelemetry

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Policies (Richtlinien) und dann Create policy (Richtlinie erstellen) aus.
3. Wählen Sie die Registerkarte JSON und kopieren Sie dann die folgende Richtlinie:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "ssm:GetParameters"
            ],
            "Resource": "*"
        }
    ]
}
```

4. Wählen Sie Richtlinie prüfen.
5. Geben Sie unter Name den Namen **`AWSDistroOpenTelemetryPolicy`** ein und wählen Sie dann Create policy (Richtlinie erstellen) aus.
6. Wählen Sie im linken Navigationsbereich Roles (Rollen) und dann Create Role (Rolle erstellen) aus.

7. Wählen Sie in der Liste der Services Elastic Container Service aus.
8. Wählen Sie unten auf der Seite Aufgabe von Elastic Container Service und dann Weiter: Berechtigungen aus.
9. Suchen Sie in der Liste der Richtlinien nach AWSDistroOpenTelemetryPolicy.
10. Aktivieren Sie das Kontrollkästchen neben AWSDistroOpenTelemetryPolicy.
11. Wählen Sie Next: Tags (Weiter: Tags) und danach Next: Review (Weiter: Prüfen) aus.
12. Geben Sie für Role name (Rollenname) den Namen **`AWSOpenTelemetryTaskRole`** ein und klicken Sie auf Create role (Rolle erstellen).

Schritt 2: Erstellen einer Aufgaben-Ausführungsrolle

Der nächste Schritt besteht darin, eine Rolle zur Aufgabenausführung für den AWS OpenTelemetry Collector zu erstellen.

Um eine Aufgabenausführungsrolle für AWS Distro zu erstellen für OpenTelemetry

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Roles (Rollen) und dann Create Role (Rolle erstellen) aus.
3. Wählen Sie in der Liste der Services Elastic Container Service aus.
4. Wählen Sie unten auf der Seite Aufgabe von Elastic Container Service und dann Weiter: Berechtigungen aus.
5. Suchen Sie in der Liste der Richtlinien nach AmazonECS TaskExecutionRolePolicy und aktivieren Sie dann das Kontrollkästchen neben TaskExecutionRolePolicyAmazonECS.
6. Suchen Sie in der Liste der Richtlinien nach CloudWatchLogsFullAccessund aktivieren Sie das Kontrollkästchen neben. CloudWatchLogsFullAccess
7. Suchen Sie in der Richtlinienliste nach AmazonSSM ReadOnlyAccess und aktivieren Sie dann das Kontrollkästchen neben AmazonSSM. ReadOnlyAccess
8. Wählen Sie Next: Tags (Weiter: Tags) und danach Next: Review (Weiter: Prüfen) aus.
9. Geben Sie für Role name (Rollenname) den Namen **`AWSOpenTelemetryTaskExecutionRole`** ein und klicken Sie auf Create role (Rolle erstellen).

Schritt 3: Erstellen einer Aufgabendefinition

Der nächste Schritt ist das Erstellen einer Aufgabendefinition.

Um eine Aufgabendefinition für Distro zu erstellen für AWS OpenTelemetry

1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/ecs/v2.
2. Wählen Sie im Navigationsbereich Task definitions (Aufgabendefinitionen) aus.
3. Wählen Sie Create new task definition (Neue Aufgabendefinition erstellen), Create new task definition (Neue Aufgabendefinition erstellen).
4. Geben Sie für Task definition family (Aufgabendefinitions-Familie) einen eindeutigen Namen für die Aufgabendefinition an.
5. Konfigurieren Sie Ihre Container und wählen Sie Weiter.
6. Wählen Sie unter Metriken und Protokollieren die Option Metrikerfassung verwenden aus.
7. Wählen Sie Next.
8. Wählen Sie Create aus.

Weitere Informationen zur Verwendung des AWS OpenTelemetry Collectors mit Amazon ECS finden Sie unter Setting up AWS Distro for OpenTelemetry Collector in Amazon Elastic Container Service.

Schritt 4: Ausführen einer Aufgabe

Im letzten Schritt wird die Aufgabe ausgeführt, die Sie erstellt haben.

Um die Aufgabe für AWS Distro auszuführen für OpenTelemetry

1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/ecs/v2.
2. Wählen Sie im linken Navigationsbereich Aufgabendefinitionen und dann die soeben erstellte Aufgabe aus.
3. Wählen Sie Aktionen, Bereitstellen, Aufgabe ausführen aus.
4. Wählen Sie Deploy (Bereitstellen), Run task (Aufgabe ausführen) aus.
5. Wählen Sie im Abschnitt Rechenoptionen unter Bestehender Cluster den gewünschten Cluster aus.
6. Wählen Sie Erstellen.
7. Als Nächstes können Sie in der CloudWatch Konsole nach den neuen Metriken suchen.
8. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
9. Wählen Sie im linken Navigationsbereich Metrics (Metriken) aus.

Sie sollten einen ECS/-Namespace ContainerInsights sehen. Wählen Sie diesen Namespace und Sie sollten acht Metriken sehen.

Bereitstellung des CloudWatch Agenten zur Erfassung von Metriken auf EC2-Instanzebene auf Amazon ECS

Um den CloudWatch Agenten zur Erfassung von Metriken auf Instanzebene aus Amazon ECS-Clustern einzusetzen, die auf einer EC2-Instance gehostet werden, verwenden Sie ein Schnellstart-Setup mit einer Standardkonfiguration oder installieren Sie den Agenten manuell, um ihn anpassen zu können.

Beide Methoden setzen voraus, dass Sie bereits mindestens einen Amazon ECS-Cluster mit einem EC2-Starttyp bereitgestellt haben und dass der CloudWatch Agent-Container Zugriff auf den Amazon EC2 Instance Metadata Service (IMDS) hat. Weitere Informationen finden Sie unter Instance-Metadaten und Benutzerdaten.

Bei diesen Methoden wird außerdem davon ausgegangen, dass Sie den installiert haben. AWS CLI Um die Befehle in den folgenden Verfahren ausführen zu können, müssen Sie außerdem bei einem Konto oder einer Rolle angemeldet sein, für das die IAM - FullAccess und FullAccessAmazonECS_-Richtlinien gelten.

Themen

- Schnelle Einrichtung mit AWS CloudFormation
- Manuelle und benutzerdefinierte Einrichtung

Schnelle Einrichtung mit AWS CloudFormation

Um das Schnell-Setup zu verwenden, geben Sie den folgenden Befehl ein, mit dem Sie den Agenten AWS CloudFormation installieren können. Ersetzen Sie *`cluster-name`* und *`cluster-region`* durch den Namen und die Region des Amazon-ECS-Clusters.

Dieser Befehl erstellt die IAM-Rollen CWAgentecs und CWAgentecs TaskRole. ExecutionRole Wenn diese Rollen bereits in Ihrem Konto vorhanden sind, verwenden Sie `ParameterKey=CreateIAMRoles,ParameterValue=False` anstelle von `ParameterKey=CreateIAMRoles,ParameterValue=True` bei der Eingabe des Befehls. Andernfalls schlägt der Befehl fehl.

```
ClusterName=cluster-name
```

```
Region=cluster-region
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/cwagent-
ecs-instance-metric/cloudformation-quickstart/cwagent-ecs-instance-metric-cfn.json
aws cloudformation create-stack --stack-name CWAgentECS-${ClusterName}-${Region} \
    --template-body file://cwagent-ecs-instance-metric-cfn.json \
    --parameters ParameterKey=ClusterName,ParameterValue=${ClusterName} \
                 ParameterKey=CreateIAMRoles,ParameterValue=True \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${Region}
```

(Alternative) Verwenden Ihrer eigenen IAM-Rollen

Wenn Sie anstelle der Rollen CWAgentecs und TaskRoleCWAgentecs Ihre eigene benutzerdefinierte ECS-Aufgabenrolle und ECS-Aufgabenausführungsrolle verwenden möchten, stellen Sie zunächst sicher, dass die Rolle, die als ExecutionRoleECS-Aufgabenrolle verwendet werden soll, angehängt ist. CloudWatchAgentServerPolicy Stellen Sie außerdem sicher, dass der Rolle, die als ECS-Aufgabenausführungsrolle verwendet werden soll, CloudWatchAgentServerPolicysowohl die Richtlinien als auch die TaskExecutionRolePolicyAmazonECS-Richtlinien angehängt sind. Geben Sie dann den folgenden Befehl ein. Ersetzen Sie im Befehl durch den *task-role-arn*ARN Ihrer benutzerdefinierten ECS-Aufgabenrolle und *execution-role-arn*ersetzen Sie ihn durch den ARN Ihrer benutzerdefinierten ECS-Aufgabenausführungsrolle.

```
ClusterName=cluster-name
Region=cluster-region
TaskRoleArn=task-role-arn
ExecutionRoleArn=execution-role-arn
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/cwagent-
ecs-instance-metric/cloudformation-quickstart/cwagent-ecs-instance-metric-cfn.json
aws cloudformation create-stack --stack-name CWAgentECS-${ClusterName}-${Region} \
    --template-body file://cwagent-ecs-instance-metric-cfn.json \
    --parameters ParameterKey=ClusterName,ParameterValue=${ClusterName} \
                 ParameterKey=TaskRoleArn,ParameterValue=${TaskRoleArn} \
                 ParameterKey=ExecutionRoleArn,ParameterValue=${ExecutionRoleArn} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${Region}
```

Fehlerbehebung bei der Schnelleinrichtung

Geben Sie den folgenden Befehl ein, um den Status des AWS CloudFormation Stacks zu überprüfen.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stacks --stack-name CWAgentECS-$ClusterName-$Region --
region $Region
```

Wenn Sie sehen, dass es sich beim `StackStatus` nicht um `CREATE_COMPLETE` oder `CREATE_IN_PROGRESS` handelt, überprüfen Sie die Stack-Ereignisse, um den Fehler zu finden. Geben Sie den folgenden Befehl ein.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stack-events --stack-name CWAgentECS-$ClusterName-$Region
 --region $Region
```

Um den Status des `cwagent`-Daemon-Services zu überprüfen, geben Sie den folgenden Befehl ein. In der Ausgabe sollten Sie sehen, dass die `runningCount` gleich der `desiredCount` im Abschnitt `deployment` ist. Wenn sie nicht gleich ist, überprüfen Sie den Abschnitt `failures` in der Ausgabe.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs describe-services --services cwagent-daemon-service --cluster $ClusterName --
region $Region
```

Sie können auch die CloudWatch Logs-Konsole verwenden, um das Agent-Log zu überprüfen. Suchen Sie nach der Protokollgruppe /ecs/ ecs-cwagent-daemon-service.

Den AWS CloudFormation Stack für den Agenten löschen CloudWatch

Wenn Sie den AWS CloudFormation Stack löschen müssen, geben Sie den folgenden Befehl ein.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation delete-stack --stack-name CWAgentECS-${ClusterName}-${Region} --
region ${Region}
```

Manuelle und benutzerdefinierte Einrichtung

Folgen Sie den Schritten in diesem Abschnitt, um den CloudWatch Agenten manuell bereitzustellen, um Metriken auf Instance-Ebene aus Ihren Amazon ECS-Clustern zu sammeln, die auf EC2-Instances gehostet werden.

Erforderliche IAM-Rollen und -Richtlinien

Zwei IAM-Rollen sind erforderlich. Sie müssen sie erstellen, wenn sie noch nicht vorhanden sind. Weitere Informationen zu diesen Rollen finden Sie unter IAM-Rollen für Aufgaben und Amazon ECS-Aufgabenausführungsrolle.

- Eine ECS-Aufgabenrolle, die vom CloudWatch Agenten verwendet wird, um Metriken zu veröffentlichen. Wenn diese Rolle bereits vorhanden ist, müssen Sie sicherstellen, dass die `CloudWatchAgentServerPolicy`-Richtlinie angehängt ist.
- Eine Rolle zur Ausführung von ECS-Aufgaben, die vom Amazon ECS-Agenten verwendet wird, um den CloudWatch Agenten zu starten. Wenn diese Rolle bereits vorhanden ist, müssen Sie sicherstellen, dass die Richtlinien `AmazonECSTaskExecutionRolePolicy` und `CloudWatchAgentServerPolicy` angehängt sind.

Wenn Sie diese Rollen noch nicht besitzen, können Sie die folgenden Befehle verwenden, um sie zu erstellen und die erforderlichen Richtlinien anzuhängen. Mit diesem ersten Befehl wird die ECS-Aufgabenrolle erstellt.

```
aws iam create-role --role-name CWAgentECSTaskRole \
    --assume-role-policy-document "{\"Version\": \"2012-10-17\",\"Statement\": [{\"Sid
\": \"\",\"Effect\": \"Allow\",\"Principal\": {\"Service\": \"ecs-tasks.amazonaws.com
\"},\"Action\": \"sts:AssumeRole\"}]}"
```

Nachdem Sie den vorherigen Befehl eingegeben haben, notieren Sie sich den Wert von `Arn` aus der Befehlsausgabe als "TaskRoleArn“. Sie müssen ihn später verwenden, wenn Sie die Aufgabendefinition erstellen. Geben Sie dann den folgenden Befehl ein, um die erforderlichen Richtlinien anzuhängen.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
CloudWatchAgentServerPolicy \
    --role-name CWAgentECSTaskRole
```

Mit diesem nächsten Befehl wird die ECS-Aufgabenausführungsrolle erstellt.

```
aws iam create-role --role-name CWAgentECSExecutionRole \
    --assume-role-policy-document "{\"Version\": \"2012-10-17\",\"Statement\": [{\"Sid
\": \"\",\"Effect\": \"Allow\",\"Principal\": {\"Service\": \"ecs-tasks.amazonaws.com
\"},\"Action\": \"sts:AssumeRole\"}]}"
```

Nachdem Sie den vorherigen Befehl eingegeben haben, notieren Sie sich den Wert von `Arn` aus der Befehlsausgabe als "ExecutionRoleArn“. Sie müssen ihn später verwenden, wenn Sie die Aufgabendefinition erstellen. Geben Sie dann die folgenden Befehle ein, um die erforderlichen Richtlinien anzuhängen.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
CloudWatchAgentServerPolicy \
    --role-name CWAgentECSExecutionRole

aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy \
    --role-name CWAgentECSExecutionRole
```

Erstellen der Aufgabendefinition und Starten des Daemon-Services

Erstellen Sie eine Aufgabendefinition und verwenden Sie sie, um den CloudWatch Agenten als Daemon-Service zu starten. Geben Sie den folgenden Befehl ein, um die Aufgabendefinition zu erstellen. Ersetzen Sie in den ersten Zeilen die Platzhalter durch die tatsächlichen Werte für Ihre Bereitstellung. *`logs-region`* ist die Region, in der sich CloudWatch Logs befindet, und *`cluster-region ist die Region`*, in der sich Ihr Cluster befindet. *`task-role-arn`*ist der Arn der ECS-Aufgabenrolle, die Sie verwenden, und *`execution-role-arn`*ist der Arn der ECS-Aufgabenausführungsrolle.

```
TaskRoleArn=task-role-arn
ExecutionRoleArn=execution-role-arn
AWSLogsRegion=logs-region
Region=cluster-region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/cwagent-
ecs-instance-metric/cwagent-ecs-instance-metric.json \
    | sed "s|{{task-role-arn}}|${TaskRoleArn}|;s|{{execution-role-arn}}|
${ExecutionRoleArn}|;s|{{awslogs-region}}|${AWSLogsRegion}|" \
    | xargs -0 aws ecs register-task-definition --region ${Region} --cli-input-json
```

Führen Sie dann den folgenden Befehl aus, um den Daemon-Service zu starten. Ersetzen Sie *`cluster-name`* und *`cluster-region`* durch den Namen und die Region des Amazon-ECS-Clusters.

> **Important**
>
> Entfernen Sie alle Strategien für Kapazitätsanbieter, bevor Sie diesen Befehl ausführen. Andernfalls funktioniert der Befehl nicht.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs create-service \
    --cluster ${ClusterName} \
    --service-name cwagent-daemon-service \
    --task-definition ecs-cwagent-daemon-service \
    --scheduling-strategy DAEMON \
    --region ${Region}
```

Wenn diese Fehlermeldung angezeigt wird, `An error occurred (InvalidParameterException) when calling the CreateService operation: Creation of service was not idempotent`, haben Sie bereits einen Daemon-Service mit dem Namen `cwagent-daemon-service` erstellt. Sie müssen diesen Service zuerst löschen, indem Sie den folgenden Befehl als Beispiel verwenden.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs delete-service \
    --cluster ${ClusterName} \
    --service cwagent-daemon-service \
    --region ${Region} \
    --force
```

(Optional) Erweiterte Konfiguration

Optional können Sie SSM verwenden, um andere Konfigurationsoptionen für den CloudWatch Agenten in Ihren Amazon ECS-Clustern anzugeben, die auf EC2-Instances gehostet werden. Es handelt sich um folgende Optionen:

- `metrics_collection_interval`— Wie oft in Sekunden sammelt der CloudWatch Agent Metriken. Der Standardwert ist 60. Der Bereich liegt zwischen 1 und 172 000.

- `endpoint_override` – (Optional) Gibt einen anderen Endpunkt an, an den Protokolle gesendet werden sollen. Dies ist sinnvoll, wenn Sie aus einem Cluster in einer VPC veröffentlichen und die Protokolldaten zu einem VPC-Endpunkt weiterleiten möchten.

  Der Wert von `endpoint_override` muss eine Zeichenkette sein, die eine URL ist.
- `force_flush_interval` – Gibt die maximale Zeitspanne in Sekunden an, in der Protokolle im Speicherpuffer verbleiben, bevor sie an den Server gesendet werden. Unabhängig von der Einstellung für dieses Feld werden die Protokolle an den Server gesendet, sobald die Größe der Protokolle im Puffer 1 MB erreicht. Der Standardwert liegt bei 5 Sekunden.
- `region` – Standardmäßig veröffentlicht der Agent Metriken in derselben Region, in der sich die Amazon-ECS-Container-Instance befindet. Um dies zu überschreiben, können Sie hier eine andere Region angeben. Beispiel: `"region" : "us-east-1"`

Im Folgenden finden Sie ein Beispiel für eine benutzerdefinierte Konfiguration:

```
{
    "agent": {
        "region": "us-east-1"
    },
    "logs": {
        "metrics_collected": {
            "ecs": {
                "metrics_collection_interval": 30
            }
        },
        "force_flush_interval": 5
    }
}
```

Um Ihre CloudWatch Agentenkonfiguration in Ihren Amazon ECS-Containern anzupassen

1. Stellen Sie sicher, dass die ReadOnlyAccessAmazonSSM-Richtlinie mit Ihrer Amazon ECS-Aufgabenausführungsrolle verknüpft ist. Dazu können Sie den folgenden Befehl eingeben. In diesem Beispiel wird davon ausgegangen, dass Ihre Amazon ECS Task Execution-Rolle ExecutionRole CWAgentecs ist. Wenn Sie eine andere Rolle verwenden, ersetzen Sie diesen Rollennamen im folgenden Befehl.

   ```
   aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
   AmazonSSMReadOnlyAccess \
   ```

```
        --role-name CWAgentECSExecutionRole
```

2. Erstellen Sie die angepasste Konfigurationsdatei ähnlich dem vorangegangenen Beispiel. Benennen Sie diese Datei `/tmp/ecs-cwagent-daemon-config.json`.
3. Führen Sie den folgenden Befehl aus, um diese Konfiguration im Parameter Store zu speichern. Ersetzen Sie *`cluster-region`* durch die Region Ihres Amazon-ECS-Clusters. Um diesen Befehl ausführen zu können, müssen Sie bei einem Benutzer oder einer Rolle angemeldet sein, für die die AmazonSSM-Richtlinie gilt. FullAccess

```
Region=cluster-region
aws ssm put-parameter \
    --name "ecs-cwagent-daemon-service" \
    --type "String" \
    --value "`cat /tmp/ecs-cwagent-daemon-config.json`" \
    --region $Region
```

4. Laden Sie die Aufgabendefinitionsdatei in eine lokale Datei herunter, z. B. `/tmp/cwagent-ecs-instance-metric.json`

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/
cwagent-ecs-instance-metric/cwagent-ecs-instance-metric.json -o /tmp/cwagent-ecs-
instance-metric.json
```

5. Ändern Sie die Aufgabendefinitionsdatei. Entfernen Sie den folgenden Abschnitt:

```
"environment": [
                {
                    "name": "USE_DEFAULT_CONFIG",
                    "value": "True"
                }
            ],
```

Ersetzen Sie diesen Abschnitt durch Folgendes:

```
"secrets": [
                {
                    "name": "CW_CONFIG_CONTENT",
                    "valueFrom": "ecs-cwagent-daemon-service"
                }
```

```
            ],
```

6. Starten Sie den Agenten als Daemon-Service neu, indem Sie die folgenden Schritte ausführen:

   a. Führen Sie den folgenden Befehl aus.

```
TaskRoleArn=task-role-arn
ExecutionRoleArn=execution-role-arn
AWSLogsRegion=logs-region
Region=cluster-region
cat /tmp/cwagent-ecs-instance-metric.json \
    | sed "s|{{task-role-arn}}|${TaskRoleArn}|;s|{{execution-role-arn}}|
${ExecutionRoleArn}|;s|{{awslogs-region}}|${AWSLogsRegion}|" \
    | xargs -0 aws ecs register-task-definition --region ${Region} --cli-input-
json
```

   b. Führen Sie den folgenden Befehl aus, um den Daemon-Service zu starten. Ersetzen Sie *cluster-name* und *cluster-region* durch den Namen und die Region des Amazon-ECS-Clusters.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs create-service \
    --cluster ${ClusterName} \
    --service-name cwagent-daemon-service \
    --task-definition ecs-cwagent-daemon-service \
    --scheduling-strategy DAEMON \
    --region ${Region}
```

   Wenn diese Fehlermeldung angezeigt wird, `An error occurred (InvalidParameterException) when calling the CreateService operation: Creation of service was not idempotent`, haben Sie bereits einen Daemon-Service mit dem Namen `cwagent-daemon-service` erstellt. Sie müssen diesen Service zuerst löschen, indem Sie den folgenden Befehl als Beispiel verwenden.

```
ClusterName=cluster-name
Region=Region
aws ecs delete-service \
    --cluster ${ClusterName} \
    --service cwagent-daemon-service \
    --region ${Region} \
```

```
    --force
```

Bereitstellung der AWS Distribution für die Erfassung von Metriken OpenTelemetry auf EC2-Instanzebene auf Amazon ECS-Clustern

Gehen Sie wie in diesem Abschnitt beschrieben vor, um mit AWS Distro for EC2-Metriken auf Instanzebene auf einem Amazon ECS-Cluster OpenTelemetry zu sammeln. Weitere Informationen zur Distribution für finden Sie unter AWS Distro for OpenTelemetry.AWS OpenTelemetry

Bei diesen Schritten wird davon ausgegangen, dass Sie bereits über einen Cluster verfügen, auf dem Amazon ECS ausgeführt wird. Dieser Cluster muss mit dem EC2-Starttyp bereitgestellt werden. Weitere Informationen zur Verwendung von AWS Distro for Open Telemetry mit Amazon ECS und zur Einrichtung eines Amazon ECS-Clusters für diesen Zweck finden Sie unter Setting up AWS Distro for OpenTelemetry Collector in Amazon Elastic Container Service for EC2-Instance-Level-Metriken.

Themen

- Schnelle Einrichtung mit AWS CloudFormation
- Manuelle und benutzerdefinierte Einrichtung

Schnelle Einrichtung mit AWS CloudFormation

Laden Sie die AWS CloudFormation Vorlagendatei für die Installation von AWS Distro for OpenTelemetry Collector für Amazon ECS auf EC2 herunter. Führen Sie den folgenden Curl-Befehl aus.

```
curl -O https://raw.githubusercontent.com/aws-observability/aws-otel-collector/main/
deployment-template/ecs/aws-otel-ec2-instance-metrics-daemon-deployment-cfn.yaml
```

Nachdem Sie die Vorlagendatei heruntergeladen haben, öffnen Sie sie und ersetzen *PATH_TO_ CloudFormation _TEMPLATE durch den Pfad, in dem Sie die Vorlagendatei* gespeichert haben. Exportieren Sie dann die folgenden Parameter und führen Sie den AWS CloudFormation Befehl aus, wie im folgenden Befehl gezeigt.

- Clustername – Name des Amazon-ECS-Clusters
- AWS_REGION – Die Region, in die die Daten gesendet werden
- PATH_TO_ CloudFormation _TEMPLATE — Der Pfad, in dem Sie die Vorlagendatei gespeichert haben. AWS CloudFormation

- command — Damit AWS Distro for OpenTelemetry Collector die Metriken auf Instance-Ebene für Amazon ECS auf Amazon EC2 sammeln kann, müssen Sie für diesen Parameter Folgendes angeben`--config=/etc/ecs/otel-instance-metrics-config.yaml`.

```
ClusterName=Cluster_Name
Region=AWS_Region
command=--config=/etc/ecs/otel-instance-metrics-config.yaml
aws cloudformation create-stack --stack-name AOCECS-${ClusterName}-${Region} \
--template-body file://PATH_TO_CloudFormation_TEMPLATE \
--parameters ParameterKey=ClusterName,ParameterValue=${ClusterName} \
ParameterKey=CreateIAMRoles,ParameterValue=True \
ParameterKey=command,ParameterValue=${command} \
--capabilities CAPABILITY_NAMED_IAM \
--region ${Region}
```

Nachdem Sie diesen Befehl ausgeführt haben, verwenden Sie die Amazon-ECS-Konsole, um festzustellen, ob die Aufgabe ausgeführt wird.

Fehlerbehebung bei der Schnelleinrichtung

Geben Sie den folgenden Befehl ein, um den Status des AWS CloudFormation Stacks zu überprüfen.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stack --stack-name AOCECS-$ClusterName-$Region --region
 $Region
```

Wenn der Wert von `StackStatus` nicht `CREATE_COMPLETE` oder `CREATE_IN_PROGRESS` ist, überprüfen Sie die Stack-Ereignisse, um den Fehler zu finden. Geben Sie den folgenden Befehl ein.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stack-events --stack-name AOCECS-$ClusterName-$Region --
region $Region
```

Um den Status des `AOCECS`-Daemon-Services zu überprüfen, geben Sie den folgenden Befehl ein. In der Ausgabe sollten Sie sehen, dass die `runningCount` gleich der `desiredCount` im Abschnitt Bereitstellung ist. Wenn sie nicht gleich ist, überprüfen Sie den Abschnitt Verletzungen in der Ausgabe.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs describe-services --services AOCECS-daemon-service --cluster $ClusterName --
region $Region
```

Sie können auch die CloudWatch Logs-Konsole verwenden, um das Agent-Log zu überprüfen. Suchen Sie nach der Protokollgruppe /aws/ecs/containerinsights/ {} /performance. ClusterName

Manuelle und benutzerdefinierte Einrichtung

Folgen Sie den Schritten in diesem Abschnitt, um die AWS Distribution manuell bereitzustellen, um Metriken auf Instance-Ebene aus Ihren Amazon ECS-Clustern OpenTelemetry zu sammeln, die auf Amazon EC2 EC2-Instances gehostet werden.

Schritt 1: Erforderliche Rollen und Richtlinien

Zwei IAM-Rollen sind erforderlich. Sie müssen sie erstellen, wenn sie noch nicht vorhanden sind. Weitere Informationen zu diesen Rollen finden Sie unter IAM-Richtlinie erstellen und IAM-Rolle erstellen.

Schritt 2: Erstellen einer Aufgabendefinition

Erstellen Sie eine Aufgabendefinition und verwenden Sie sie, um AWS Distro for OpenTelemetry als Daemon-Service zu starten.

Um die Aufgabendefinitionsvorlage zum Erstellen der Aufgabendefinition zu verwenden, folgen Sie den Anweisungen unter ECS EC2-Aufgabendefinition für EC2-Instance mit oTEL Collector erstellen. AWS

Um die Amazon ECS-Konsole zum Erstellen der Aufgabendefinition zu verwenden, folgen Sie den Anweisungen unter Installieren von OTel Collector AWS , indem Sie die Aufgabendefinition über die AWS Konsole für Amazon ECS EC2-Instance-Metriken erstellen.

Schritt 3: Starten des Daemon-Services

Um die AWS Distribution for OpenTelemetry als Daemon-Service zu starten, folgen Sie den Anweisungen unter Führen Sie Ihre Aufgabe auf dem Amazon Elastic Container Service (Amazon ECS) mithilfe des Daemon-Service aus.

(Optional) Erweiterte Konfiguration

Optional können Sie SSM verwenden, um andere Konfigurationsoptionen für die AWS Distribution für OpenTelemetry Ihre Amazon ECS-Cluster anzugeben, die auf Amazon EC2 EC2-Instances gehostet werden. Weitere Informationen zum Erstellen einer Konfigurationsdatei finden Sie unter Benutzerdefinierte Konfiguration. OpenTelemetry Weitere Informationen zu den Optionen, die Sie in der Konfigurationsdatei verwenden können, finden Sie unter AWS Container Insights Receiver.

Richten Sie FireLens das Senden von Protokollen an CloudWatch Logs ein

FireLens for Amazon ECS ermöglicht es Ihnen, Aufgabendefinitionsparameter zu verwenden, um CloudWatch Protokolle zur Protokollspeicherung und Analyse an Amazon Logs weiterzuleiten. FireLens funktioniert mit Fluent Bit und Fluentd. Wir stellen ein Bild AWS für Fluent Bit zur Verfügung, oder Sie können Ihr eigenes Fluent Bit- oder Fluentd-Bild verwenden. Das Erstellen von Amazon ECS-Aufgabendefinitionen mit einer FireLens Konfiguration wird mithilfe der AWS SDKs AWS CLI, und AWS Management Console unterstützt. Weitere Informationen zu CloudWatch Logs finden Sie unter Was sind CloudWatch Logs? .

Bei der Verwendung FireLens für Amazon ECS sind wichtige Überlegungen zu beachten. Weitere Informationen finden Sie unter Überlegungen.

Informationen zu den AWS for Fluent Bit-Bildern finden Sie unter Verwenden des AWS for Fluent Bit-Images.

Informationen zum Erstellen einer Aufgabendefinition, die eine FireLens Konfiguration verwendet, finden Sie unter Aufgabendefinition erstellen, die eine FireLens Konfiguration verwendet.

Beispiel

Das folgende Beispiel für eine Aufgabendefinition zeigt, wie eine Protokollkonfiguration angegeben wird, die Protokolle an eine Protokollgruppe „ CloudWatch Logs“ weiterleitet. Weitere Informationen finden Sie unter Was ist Amazon CloudWatch Logs? im Amazon CloudWatch Logs-Benutzerhandbuch.

Geben Sie in den Protokollkonfigurationsoptionen den Namen der Protokollgruppe und die Region an, in der sie vorhanden ist. Geben Sie `"auto_create_group":"true"` an, damit Fluent Bit die Protokollgruppe in Ihrem Namen erstellt. Sie können auch die Aufgaben-ID als ein Protokoll-Stream-Präfix angeben, das beim Filtern unterstützt. Weitere Informationen finden Sie unter Fluent Bit Plugin for CloudWatch Logs.

```
{
```

```
 "family": "firelens-example-cloudwatch",
 "taskRoleArn": "arn:aws:iam::123456789012:role/ecs_task_iam_role",
 "containerDefinitions": [
  {
   "essential": true,
   "image": "906394416424.dkr.ecr.us-west-2.amazonaws.com/aws-for-fluent-bit:latest",
   "name": "log_router",
   "firelensConfiguration": {
    "type": "fluentbit"
   },
   "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
     "awslogs-group": "firelens-container",
     "awslogs-region": "us-west-2",
     "awslogs-create-group": "true",
     "awslogs-stream-prefix": "firelens"
    }
   },
   "memoryReservation": 50
   },
   {
    "essential": true,
    "image": "nginx",
    "name": "app",
    "logConfiguration": {
     "logDriver":"awsfirelens",
     "options": {
     "Name": "cloudwatch",
     "region": "us-west-2",
     "log_key": "log",
                                "log_group_name": "/aws/ecs/containerinsights/
$(ecs_cluster)/application",
     "auto_create_group": "true",
     "log_stream_name": "$(ecs_task_id)"
    }
   },
   "memoryReservation": 100
  }
 ]
}
```

Einrichten von Container Insights in Amazon EKS und Kubernetes

Container Insights wird auf Amazon-EKS-Versionen 1.23 und höher unterstützt. Die Schnellstart-Installationsmethode wird nur in den Versionen 1.24 und höher unterstützt.

Der allgemeine Prozess zum Einrichten von Container Insights in Amazon EKS oder Kubernetes gestaltet sich wie folgt:

1. Stellen Sie sicher, dass Sie die erforderlichen Voraussetzungen erfüllen.
2. Richten Sie das Amazon CloudWatch Observability EKS-Add-on, den CloudWatch Agenten oder die AWS Distribution für Ihren Cluster ein, OpenTelemetry an den Metriken gesendet werden sollen. CloudWatch

> **Note**
>
> Um Container Insights mit erweiterter Observability für Amazon EKS zu verwenden, müssen Sie das Amazon CloudWatch Observability EKS-Add-on oder den CloudWatch Agenten verwenden. Weitere Informationen zu dieser Version von Container Insights finden Sie unter Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS. Um Container Insights mit Fargate verwenden zu können, müssen Sie AWS Distro for verwenden. OpenTelemetry Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS wird auf Fargate nicht unterstützt.

> **Note**
>
> Container Insights unterstützt jetzt Windows-Worker-Knoten in einem Amazon EKS-Cluster. Container Insights mit verbesserter Observability für Amazon EKS wird auch unter Windows unterstützt. Informationen zur Aktivierung von Container Insights unter Windows finden Sie unterBei Verwendung des CloudWatch Agenten mit aktivierter erweiterter Observability von Container Insights.

Richten Sie Fluent Bit oder Fluentd ein, um Logs an Logs zu senden. CloudWatch (Dies ist standardmäßig aktiviert, wenn Sie das Amazon CloudWatch Observability EKS-Add-on installieren.)

Sie können diese Schritte gleichzeitig als Teil der Schnellstarteinrichtung ausführen, wenn Sie den CloudWatch Agenten verwenden, oder sie separat ausführen.

3. (Optional) Richten Sie die Protokollierungauf der Amazon-EKS-Steuerebene ein.
4. (Optional) Richten Sie den CloudWatch Agenten als StatsD-Endpunkt auf dem Cluster ein, an den StatsD-Metriken gesendet werden sollen. CloudWatch
5. (Optional) Aktivieren von App Mesh Envoy Access Logs.

In der Originalversion von Container Insights werden erfasste Metriken und aufgenommene Protokolle als benutzerdefinierte Metriken berechnet. Bei Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS werden die Container-Insights-Metriken und -Protokolle pro Beobachtung abgerechnet, anstatt pro gespeicherter Metrik oder aufgenommenem Protokoll. Weitere Informationen zur CloudWatch Preisgestaltung finden Sie unter CloudWatch Amazon-Preise.

Themen

- Überprüfen Sie die -Voraussetzungen.
- Bei Verwendung des CloudWatch Agenten mit aktivierter erweiterter Observability von Container Insights
- Verwenden Sie AWS Distro für OpenTelemetry
- Protokolle an Logs senden CloudWatch
- Aktualisieren oder Löschen von Container-Insights auf Amazon EKS und Kubernetes

Überprüfen Sie die -Voraussetzungen.

Bevor Sie Container Insights in Amazon EKS oder Kubernetes installieren, überprüfen Sie Folgendes: Diese Voraussetzungen gelten unabhängig davon, ob Sie den CloudWatch Agenten oder die AWS Distribution for verwenden OpenTelemetry , um Container Insights auf Amazon EKS-Clustern einzurichten.

- Sie besitzen einen funktionsfähigen Amazon EKS oder Kubernetes-Cluster mit angefügten Knoten in einer der Regionen, die Container Insights für Amazon EKS und Kubernetes unterstützen. Eine Liste der unterstützten Regionen finden Sie unter Container Insights.
- Sie haben `kubectl` installiert und es wird ausgeführt. Weitere Informationen finden Sie unter Installieren von `kubectl` im Amazon-EKS-Benutzerhandbuch.

- Wenn Sie Kubernetes, das auf läuft, AWS anstelle von Amazon EKS verwenden, sind auch die folgenden Voraussetzungen erforderlich:
  - Stellen Sie sicher, dass für Ihren Kubernetes-Cluster rollenbasierte Zugriffskontrolle (RBAC) aktiviert ist. Weitere Informationen finden Sie unter Using RBAC Authorization in der Kubernetes-Dokumentation.
  - Für Ihr Kubelet ist der Webhook-Autorisierungsmodus aktiviert. Weitere Informationen finden Sie unter Kubelet authentication/authorization in der Kubernetes-Dokumentation.

Sie müssen auch IAM-Berechtigungen gewähren, damit Ihre Amazon EKS-Worker-Knoten Metriken und Protokolle an CloudWatch sie senden können. Es gibt zwei Möglichkeiten dafür:

- Fügen Sie eine Richtlinie an die IAM-Rolle Ihrer Worker-Knoten an. Dies funktioniert sowohl für Amazon-EKS-Cluster als auch für andere Kubernetes-Cluster.
- Verwenden Sie eine IAM-Rolle für Service-Konten für den Cluster und fügen Sie die Richtlinie an diese Rolle an. Dies funktioniert nur für Amazon-EKS-Cluster.

Die erste Option gewährt Berechtigungen CloudWatch für den gesamten Knoten, während die Verwendung einer IAM-Rolle für das Dienstkonto nur CloudWatch Zugriff auf die entsprechenden Daemonset-Pods gewährt.

Anfügen einer Richtlinie an die IAM-Rolle der Worker-Knoten

Führen Sie die folgenden Schritte aus, um die Richtlinie an die IAM-Rolle Ihrer Worker-Knoten anzufügen. Dies funktioniert sowohl für Amazon-EKS-Cluster als auch für Kubernetes-Cluster außerhalb von Amazon EKS.

So fügen Sie die erforderliche Richtlinie an die IAM-Rolle für Ihre Worker-Knoten an

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie eine der Workerknoten-Instances und danach die IAM-Rolle in der Beschreibung aus.
3. Wählen Sie auf der Seite der IAM-Rolle die Option Richtlinien anfügen.
4. Aktivieren Sie in der Liste der Richtlinien das Kontrollkästchen neben. CloudWatchAgentServerPolicy Verwenden Sie ggf. das Suchfeld, um diese Richtlinie zu finden.
5. Wählen Sie Richtlinien anfügen.

Wenn Sie einen Kubernetes-Cluster außerhalb von Amazon EKS ausführen, ist Ihren Workerknoten möglicherweise noch keine IAM-Rolle zugeordnet. In diesem Fall müssen Sie der Instance zuerst eine IAM-Rolle zuweisen und die Richtlinie dann, wie in den vorherigen Schritten beschrieben, hinzufügen. Weitere Informationen zum Anhängen einer Rolle an eine Instance finden Sie unter Anhängen einer IAM-Rolle an eine Instance im Amazon-EC2-Benutzerhandbuch für Windows-Instances.

Wenn Sie einen Kubernetes-Cluster außerhalb von Amazon EKS ausführen und EBS-Volume-IDs in den Metriken erfassen möchten, müssen Sie der IAM-Rolle, die der Instance zugeordnet ist, eine weitere Richtlinie hinzufügen. Fügen Sie Folgendes als eingebundenen Richtlinie hinzu. Informationen finden Sie im Abschnitt Hinzufügen und Entfernen von IAM-Identitätsberechtigungen im IAM-Benutzerhandbuch.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeVolumes"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Verwenden einer IAM-Service-Kontorolle

Diese Methode funktioniert nur für Amazon-EKS-Cluster.

Um die Erlaubnis zur CloudWatch Verwendung einer IAM-Dienstkontorolle zu erteilen

1. Aktivieren Sie die IAM-Rollen für Service-Konten in Ihrem Cluster, falls Sie dies noch nicht getan haben. Weitere Informationen finden Sie unter Aktivieren von IAM-Rollen für Service-Konten in Ihrem Cluster.
2. Konfigurieren Sie das Servicekonto für die Nutzung einer IAM-Rolle, sofern Sie das noch nicht getan haben. Weitere Informationen finden Sie unter Konfigurieren des Kubernetes-Servicekontos zur Übernahme einer IAM-Rolle.

Wenn Sie die Rolle erstellen, fügen Sie der Rolle zusätzlich zu der Richtlinie, die Sie für die Rolle erstellen, auch die CloudWatchAgentServerPolicyIAM-Richtlinie hinzu. Außerdem sollte das zugehörige Kubernetes-Dienstkonto, das mit dieser Rolle verknüpft ist, im `amazon-cloudwatch` Namespace erstellt werden, wo die Daemonsets CloudWatch und Fluent Bit in den nächsten Schritten bereitgestellt werden

3. Ordnen Sie die IAM-Rolle einem Service-Konto in Ihrem Cluster zu, falls Sie dies noch nicht getan haben. Weitere Informationen finden Sie unter Konfigurieren des Kubernetes-Servicekontos zur Übernahme einer IAM-Rolle.

Bei Verwendung des CloudWatch Agenten mit aktivierter erweiterter Observability von Container Insights

Verwenden Sie die Anweisungen in einem der folgenden Abschnitte, um Container Insights auf einem Amazon EKS-Cluster oder Kubernetes-Cluster mithilfe des CloudWatch Agenten einzurichten. Die Schnellstartanleitung wird nur von Amazon-EKS-Versionen 1.24 und höher unterstützt.

> **Note**
>
> Sie können Container Insights installieren, indem Sie den Anweisungen in einem der folgenden Abschnitte folgen. Sie müssen nicht alle drei Anweisungen befolgen.

Themen

- Installieren Sie das Amazon CloudWatch Observability EKS-Add-on
- Schnellstarteinrichtung für Container Insights auf Amazon EKS und Kubernetes
- Richten Sie den CloudWatch Agenten für die Erfassung von Cluster-Metriken ein

Installieren Sie das Amazon CloudWatch Observability EKS-Add-on

Sie können das Amazon-EKS-Add-On verwenden, um Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS zu installieren. Das Add-on installiert den CloudWatch Agenten zum Senden von Infrastrukturmetriken aus dem Cluster, installiert Fluent Bit zum Senden von Container-Protokollen und ermöglicht CloudWatch Application Signals auch das Senden von Telemetriedaten zur Anwendungsleistung.

Wenn Sie das Amazon EKS-Add-on Version 1.5.0 oder höher verwenden, ist Container Insights sowohl auf Linux- als auch auf Windows-Worker-Knoten im Cluster aktiviert. Derzeit wird Application Signals unter Windows in Amazon EKS nicht unterstützt.

Das Amazon-EKS-Add-On wird nicht für Cluster unterstützt, auf denen Kubernetes statt Amazon EKS ausgeführt wird.

Weitere Informationen zum Amazon CloudWatch Observability EKS-Add-on finden Sie unterInstallieren Sie den CloudWatch Agenten mithilfe des Amazon CloudWatch Observability EKS-Add-ons.

So installieren Sie das Amazon CloudWatch Observability EKS-Add-on

1. Richten Sie zunächst die erforderlichen Berechtigungen ein, indem Sie die CloudWatchAgentServerPolicyIAM-Richtlinie an Ihre Worker-Knoten anhängen. Geben Sie dazu den folgenden Befehl ein. *my-worker-node-role*Ersetzen Sie sie durch die IAM-Rolle, die von Ihren Kubernetes-Worker-Knoten verwendet wird.

   ```
   aws iam attach-role-policy \
   --role-name my-worker-node-role \
   --policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
   ```

2. Geben Sie den folgenden Befehl ein, um das Add-On zu installieren:

   ```
   aws eks create-addon --cluster-name my-cluster-name --addon-name amazon-cloudwatch-observability
   ```

Schnellstarteinrichtung für Container Insights auf Amazon EKS und Kubernetes

> **Important**
>
> Wenn Sie Container Insights auf einem Amazon EKS-Cluster installieren, empfehlen wir, das Amazon CloudWatch Observability EKS-Add-on für die Installation zu verwenden, anstatt die Anweisungen in diesem Abschnitt zu verwenden. Um beschleunigte Computernetzwerke abzurufen, müssen Sie außerdem das Amazon CloudWatch Observability EKS-Add-on verwenden. Weitere Informationen und Anweisungen finden Sie unter Installieren Sie das Amazon CloudWatch Observability EKS-Add-on.

Um die Einrichtung von Container Insights abzuschließen, befolgen Sie die Quick Start-Anweisungen in diesem Abschnitt. Wenn Sie in einem Amazon-EKS-Cluster installieren und die Anweisungen in diesem Abschnitt am oder nach dem 6. November 2023 verwenden, installieren Sie Container Insights mit erweiterter Beobachtbarkeit für Amazon EKS in diesem Cluster.

> **Important**
>
> Bevor Sie die Schritte in diesem Abschnitt ausführen, müssen Sie die Voraussetzungen einschließlich IAM-Berechtigungen überprüft haben. Weitere Informationen finden Sie unter Überprüfen Sie die -Voraussetzungen..

Alternativ können Sie stattdessen die Anweisungen in den folgenden beiden Abschnitten befolgen, Richten Sie den CloudWatch Agenten für die Erfassung von Cluster-Metriken ein und Protokolle an Logs senden CloudWatch . In diesen Abschnitten finden Sie weitere Konfigurationsdetails zur Funktionsweise des CloudWatch Agenten mit Amazon EKS und Kubernetes. Sie müssen jedoch weitere Installationsschritte ausführen.

In der Originalversion von Container Insights werden erfasste Metriken und aufgenommene Protokolle als benutzerdefinierte Metriken berechnet. Bei Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS werden die Container-Insights-Metriken und -Protokolle pro Beobachtung abgerechnet, anstatt pro gespeicherter Metrik oder aufgenommenem Protokoll. Weitere Informationen zur CloudWatch Preisgestaltung finden Sie unter CloudWatchAmazon-Preise.

> **Note**
>
> Amazon hat nun Fluent Bit als Standardprotokolllösung für Container Insights mit erheblichen Leistungssteigerungen eingeführt. Wir empfehlen Ihnen, Fluent Bit anstelle von Fluentd zu verwenden.

Schnellstart mit dem CloudWatch Agenten, Operator und Fluent Bit

Es gibt zwei Konfigurationen für Fluent Bit: Eine optimierte Version und eine Version, die eine ähnliche Erfahrung wie Fluentd bietet. Die Schnellstart-Konfiguration verwendet die optimierte Version. Weitere Details zur Fluentd-kompatiblen Konfiguration finden Sie unter Richten Sie Fluent Bit ein, um Protokolle an Logs DaemonSet zu senden CloudWatch .

Der CloudWatch Agent-Operator ist ein zusätzlicher Container, der in einem Amazon EKS-Cluster installiert wird. Er ist dem OpenTelemetry Operator für Kubernetes nachempfunden. Der Betreiber verwaltet den Lebenszyklus von Kubernetes-Ressourcen in einem Cluster. Es installiert den CloudWatch Agenten, den DCGM Exporter (NVIDIA) und den AWS Neuron Monitor auf einem Amazon EKS-Cluster und verwaltet sie. Fluent Bit und der CloudWatch Agent for Windows werden direkt in einem Amazon EKS-Cluster installiert, ohne dass der Betreiber sie verwaltet.

Für eine sicherere und funktionsreichere Zertifizierungsstellenlösung benötigt der CloudWatch Agent-Betreiber cert-manager, eine weit verbreitete Lösung für die Verwaltung von TLS-Zertifikaten in Kubernetes. Die Verwendung von cert-manager vereinfacht den Prozess der Beschaffung, Erneuerung, Verwaltung und Verwendung dieser Zertifikate. Es stellt sicher, dass Zertifikate gültig und aktuell sind, und versucht, Zertifikate zu einem konfigurierten Zeitpunkt vor Ablauf zu erneuern. cert-manager erleichtert auch die Ausstellung von Zertifikaten aus einer Vielzahl unterstützter Quellen, einschließlich Certificate Manager Private AWS Certificate Authority.

So stellen Sie Container Insights mithilfe des Schnellstarts bereit

1. Installieren Sie cert-manager, falls er nicht bereits im Cluster installiert ist. Weitere Informationen finden Sie unter Installation von cert-manager.
2. Installieren Sie die benutzerdefinierten Ressourcendefinitionen (CRD), indem Sie den folgenden Befehl eingeben.

   ```
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
   insights/main/k8s-quickstart/cwagent-custom-resource-definitions.yaml | kubectl
    apply --server-side -f -
   ```

3. Installieren Sie den Operator, indem Sie den folgenden Befehl eingeben. *my-cluster-name*Ersetzen Sie es durch den Namen Ihres Amazon EKS- oder Kubernetes-Clusters und *my-cluster-region*ersetzen Sie es durch den Namen der Region, in der die Protokolle veröffentlicht werden. Wir empfehlen, dass Sie dieselbe Region verwenden, in der Ihr Cluster bereitgestellt wird, um die Kosten für AWS ausgehende Datenübertragungen zu reduzieren.

   ```
   ClusterName=my-cluster-name
   RegionName=my-cluster-region
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
   container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
   {{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
    apply -f -
   ```

Um beispielsweise Container Insights auf dem Cluster mit dem Namen `MyCluster` bereitzustellen und die Protokolle und Metriken in USA West (Oregon) zu veröffentlichen, geben Sie den folgenden Befehl ein.

```
ClusterName='MyCluster'
RegionName='us-west-2'
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 apply -f -
```

Migration von Container Insights

Wenn Sie Container Insights bereits in einem Amazon EKS-Cluster konfiguriert haben und zu Container Insights mit verbesserter Observability für Amazon EKS migrieren möchten, finden Sie weitere Informationen unter Upgrade auf Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS

Löschen von Container Insights

Wenn Sie Container Insights entfernen möchten, nachdem Sie das Schnellstart-Setup verwendet haben, geben Sie die folgenden Befehle ein.

```
ClusterName=my-cluster-name
RegionName=my-cluster-region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 delete -f -
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/main/k8s-quickstart/cwagent-custom-resource-definitions.yaml | kubectl delete
 -f -
```

Richten Sie den CloudWatch Agenten für die Erfassung von Cluster-Metriken ein

> **Important**
>
> Wenn Sie Container Insights auf einem Amazon EKS-Cluster installieren, empfehlen wir Ihnen, das Amazon CloudWatch Observability EKS-Add-on für die Installation zu verwenden, anstatt die Anweisungen in diesem Abschnitt zu verwenden. Weitere Informationen und

Anweisungen finden Sie unter Installieren Sie das Amazon CloudWatch Observability EKS-Add-on.

Um Container Insights zum Sammeln von Metriken einzurichten, können Sie die Schritte unter Schnellstarteinrichtung für Container Insights auf Amazon EKS und Kubernetes oder die Schritte in diesem Abschnitt befolgen. In den folgenden Schritten richten Sie den CloudWatch Agenten so ein, dass er Metriken aus Ihren Clustern sammeln kann.

Wenn Sie in einem Amazon-EKS-Cluster installieren und die Anweisungen in diesem Abschnitt am oder nach dem 6. November 2023 verwenden, installieren Sie Container Insights mit erweiterter Beobachtbarkeit für Amazon EKS in diesem Cluster.

Schritt 1: Erstellen Sie einen Namespace für CloudWatch

Verwenden Sie den folgenden Schritt, um einen angeforderten Kubernetes-Namespace zu erstellen. `amazon-cloudwatch` CloudWatch Sie können diesen Schritt überspringen, wenn Sie diesen Namespace bereits erstellt haben.

Um einen Namespace zu erstellen für CloudWatch

- Geben Sie den folgenden Befehl ein.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
daemonset/container-insights-monitoring/cloudwatch-namespace.yaml
```

Schritt 2: Erstellen eines Servicekontos im Cluster

Verwenden Sie den folgenden Schritt, um ein Dienstkonto für den CloudWatch Agenten zu erstellen, falls Sie noch keines haben.

Um ein Dienstkonto für den CloudWatch Agenten zu erstellen

- Geben Sie den folgenden Befehl ein.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
daemonset/container-insights-monitoring/cwagent/cwagent-serviceaccount.yaml
```

Wenn Sie die vorherigen Schritte nicht befolgt haben, aber bereits über ein Dienstkonto für den CloudWatch Agenten verfügen, den Sie verwenden möchten, müssen Sie sicherstellen, dass für ihn die folgenden Regeln gelten. Außerdem müssen Sie in der restlichen Schritten der Container Insights-Installation statt `cloudwatch-agent` den Namen des betreffenden Servicekontos verwenden.

```
rules:
  - apiGroups: [""]
    resources: ["pods", "nodes", "endpoints"]
    verbs: ["list", "watch"]
  - apiGroups: [ "" ]
    resources: [ "services" ]
    verbs: [ "list", "watch" ]
  - apiGroups: ["apps"]
    resources: ["replicasets", "daemonsets", "deployments", "statefulsets"]
    verbs: ["list", "watch"]
  - apiGroups: ["batch"]
    resources: ["jobs"]
    verbs: ["list", "watch"]
  - apiGroups: [""]
    resources: ["nodes/proxy"]
    verbs: ["get"]
  - apiGroups: [""]
    resources: ["nodes/stats", "configmaps", "events"]
    verbs: ["create", "get"]
  - apiGroups: [""]
    resources: ["configmaps"]
    resourceNames: ["cwagent-clusterleader"]
    verbs: ["get","update"]
  - nonResourceURLs: ["/metrics"]
    verbs: ["get", "list", "watch"]
```

Schritt 3: Erstellen Sie ein ConfigMap für den CloudWatch Agenten

Gehen Sie wie folgt vor, um eine ConfigMap für den CloudWatch Agenten zu erstellen.

Um eine ConfigMap für den CloudWatch Agenten zu erstellen

1. Laden Sie das ConfigMap YAML auf Ihren `kubectl` Client-Host herunter, indem Sie den folgenden Befehl ausführen:

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/
container-insights-monitoring/cwagent/cwagent-configmap.yaml
```

2. Bearbeiten Sie die heruntergeladene YAML-Datei wie folgt:

   - cluster_name – Ersetzen Sie im Abschnitt `kubernetes` den Platzhalter `{{cluster_name}}` durch den Namen Ihres Clusters. Entfernen Sie die Zeichen `{{}}`. Wenn Sie einen Amazon-EKS-Cluster verwenden, können Sie das Feld `"cluster_name"` und den Wert löschen. Wenn Sie dies tun, erkennt der CloudWatch Agent den Clusternamen anhand der Amazon EC2-Tags.

3. (Optional) Nehmen Sie auf der ConfigMap Grundlage Ihrer Überwachungsanforderungen weitere Änderungen an der vor, und zwar wie folgt:

   - metrics_collection_interval – Im Abschnitt `kubernetes` können Sie angeben, wie oft der Agent Metriken sammelt. Standardmäßig ist ein Zeitraum von 60 Sekunden festgelegt. Das standardmäßige cadvisor-Erfassungsintervall in Kubelet beträgt 15 Sekunden. Stellen Sie für diesen Wert also nicht weniger als 15 Sekunden ein.
   - endpoint_override — In `logs` diesem Abschnitt können Sie den Endpunkt CloudWatch Logs angeben, wenn Sie den Standardendpunkt überschreiben möchten. Dies ist sinnvoll, wenn Sie aus einem Cluster in einer VPC veröffentlichen und die Daten zu einem VPC-Endpunkt weiterleiten möchten.
   - force_flush_interval — In `logs` diesem Abschnitt können Sie das Intervall angeben, in dem Protokollereignisse gebündelt werden, bevor sie in Logs veröffentlicht werden. CloudWatch Standardmäßig ist ein Zeitraum von 5 Sekunden festgelegt.
   - Region – Standardmäßig veröffentlicht der Agent Metriken in der Region, in der sich der Workerknoten befindet. Um diese Einstellung zu überschreiben, können Sie ein `region`-Feld zum Abschnitt `agent` hinzufügen. Beispiel: `"region":"us-west-2"`.
   - statsd-Abschnitt — Wenn Sie möchten, dass der CloudWatch Logs-Agent auch als StatsD-Listener in jedem Worker-Knoten Ihres Clusters ausgeführt wird, können Sie dem `statsd` `metrics` Abschnitt einen Abschnitt hinzufügen, wie im folgenden Beispiel. Weitere Informationen zu anderen StatsD-Optionen für diesen Abschnitt finden Sie unter Abrufen benutzerdefinierter Metriken mit StatsD .

   ```
   "metrics": {
     "metrics_collected": {
   ```

```
    "statsd": {
      "service_address":":8125"
    }
  }
}
```

Ein vollständiges Beispiel für den JSON-Abschnitt lautet wie folgt.

```
{
    "agent": {
        "region": "us-east-1"
    },
    "logs": {
        "metrics_collected": {
            "kubernetes": {
                "cluster_name": "MyCluster",
                "metrics_collection_interval": 60
            }
        },
        "force_flush_interval": 5,
        "endpoint_override": "logs.us-east-1.amazonaws.com"
    },
    "metrics": {
        "metrics_collected": {
            "statsd": {
                "service_address": ":8125"
            }
        }
    }
}
```

4. Erstellen Sie den ConfigMap im Cluster, indem Sie den folgenden Befehl ausführen.

```
kubectl apply -f cwagent-configmap.yaml
```

Schritt 4: Stellen Sie den CloudWatch Agenten als DaemonSet

Gehen Sie wie folgt vor, um die Installation des CloudWatch Agenten abzuschließen und mit der Erfassung von Container-Metriken zu beginnen.

Um den CloudWatch Agenten als zu installieren DaemonSet

1. • Wenn Sie StatsD nicht auf dem Cluster verwenden möchten, geben Sie den folgenden Befehl ein.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
deployment-mode/daemonset/container-insights-monitoring/cwagent/cwagent-
daemonset.yaml
```

   • Wenn Sie StatsD verwenden möchten, führen Sie die folgenden Schritte aus:

   a. Laden Sie das DaemonSet YAML auf Ihren `kubectl` Client-Host herunter, indem Sie den folgenden Befehl ausführen.

```
curl -O  https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-
mode/daemonset/container-insights-monitoring/cwagent/cwagent-daemonset.yaml
```

   b. Heben Sie die Kommentierung des `port`-Abschnitts in der `cwagent-daemonset.yaml`-Datei wie folgt auf:

```
ports:
  - containerPort: 8125
    hostPort: 8125
    protocol: UDP
```

   c. Stellen Sie den CloudWatch Agenten in Ihrem Cluster bereit, indem Sie den folgenden Befehl ausführen.

```
kubectl apply -f cwagent-daemonset.yaml
```

   d. Stellen Sie den CloudWatch Agenten auf Windows-Knoten in Ihrem Cluster bereit, indem Sie den folgenden Befehl ausführen. Der StatsD-Listener wird auf dem CloudWatch Agenten unter Windows nicht unterstützt.

```
kubectl apply -f cwagent-daemonset-windows.yaml
```

2. Überprüfen Sie, ob der Agent bereitgestellt wird, indem Sie den folgenden Befehl ausführen.

```
kubectl get pods -n amazon-cloudwatch
```

Wenn der Vorgang abgeschlossen ist, erstellt der CloudWatch Agent eine Protokollgruppe mit dem Namen `/aws/containerinsights/Cluster_Name/performance` und sendet die Leistungsprotokollereignisse an diese Protokollgruppe. Wenn Sie den Agenten als auch StatsD-Listener einrichten, überwacht der Agent Port 8125 mit der IP-Adresse des Knotens, auf dem der Anwendung-Pod geplant ist, auch auf StatsD-Metriken.

Fehlerbehebung

Wenn der Agent nicht korrekt bereitgestellt wird, führen Sie die folgenden Schritte aus:

- Führen Sie den folgenden Befehl aus, um die Liste der Pods zu erhalten.

```
kubectl get pods -n amazon-cloudwatch
```

- Führen Sie den folgenden Befehl aus und überprüfen Sie die Ereignisse am unteren Rand der Ausgabe.

```
kubectl describe pod pod-name -n amazon-cloudwatch
```

- Führen Sie den folgenden Befehl aus, um die Protokolle zu überprüfen.

```
kubectl logs pod-name  -n amazon-cloudwatch
```

Verwenden Sie AWS Distro für OpenTelemetry

Sie können Container Insights so einrichten, dass Metriken aus Amazon EKS-Clustern gesammelt werden, indem Sie AWS Distro for OpenTelemetry Collector verwenden. Weitere Informationen zur Distribution für finden Sie unter AWS AWS Distro for OpenTelemetry. OpenTelemetry

> **Important**
>
> Wenn Sie mit AWS Distro for installieren OpenTelemetry, installieren Sie Container Insights, erhalten aber Container Insights mit erweiterter Observability für Amazon EKS nicht. Sie werden die detaillierten Metriken, die in Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS unterstützt werden, nicht erfassen.

Wie Sie Container Insights einrichten, hängt davon ab, ob der Cluster auf Amazon-EC2-Instances oder auf AWS Fargate (Fargate) gehostet wird.

Amazon-EKS-Cluster, die auf Amazon EC2 gehostet werden

Wenn Sie dies nicht bereits getan haben, stellen Sie sicher, dass Sie die Voraussetzungen einschließlich der erforderlichen IAM-Rollen erfüllt haben. Weitere Informationen finden Sie unter Überprüfen Sie die -Voraussetzungen..

Amazon stellt ein Helm-Chart bereit, mit dem Sie die Überwachung von Amazon EKS (Elastic Kubernetes Service) auf Amazon EC2 einrichten können. Diese Überwachung verwendet den AWS Distro for OpenTelemetry (ADOT) Collector für Metriken und Fluent Bit für Protokolle. Daher ist das Helm-Diagramm für Kunden nützlich, die Amazon EKS auf Amazon EC2 verwenden und Metriken und Protokolle sammeln möchten, um sie an CloudWatch Container Insights zu senden. Weitere Informationen zu diesem Helm-Diagramm finden Sie unter ADOT-Helm-Diagramm für EKS zu EC2-Metriken und Protokollen in Amazon CloudWatch Container Insights.

Alternativ können Sie die Anweisungen im Rest dieses Abschnitts verwenden.

Stellen Sie zunächst die AWS Distribution für OpenTelemetry Collector als bereit, DaemonSet indem Sie den folgenden Befehl eingeben.

```
curl https://raw.githubusercontent.com/aws-observability/aws-otel-collector/main/
deployment-template/eks/otel-container-insights-infra.yaml |
kubectl apply -f -
```

Verwenden Sie den folgenden Befehl, um zu bestätigen, dass der Collector ausgeführt wird:

```
kubectl get pods -l name=aws-otel-eks-ci -n aws-otel-eks
```

Wenn die Ausgabe dieses Befehls mehrere Pods im `Running`-Zustand enthält, wird der Collector ausgeführt und erfasst Metriken vom Cluster. Der Collector erstellt eine Protokollgruppe namens `aws/containerinsights/`*`cluster-name`*`/performance` und sendet die Leistungsprotokollereignisse an sie.

Informationen darüber, wie Sie Ihre Container Insights-Metriken in einsehen können CloudWatch, finden Sie unterAnzeigen von Container-Insights-Metriken.

AWS hat auch Dokumentation GitHub zu diesem Szenario bereitgestellt. Informationen zum Anpassen der von Container Insights veröffentlichten Metriken und Protokolle finden Sie unter https://aws-otel.github.io/docs/getting-started/container-insights/eks-infra.

Auf Fargate gehostete Amazon-EKS-Cluster

Anweisungen zur Konfiguration und Bereitstellung eines ADOT Collectors zum Sammeln von Systemmetriken von Workloads, die in einem Amazon EKS-Cluster auf Fargate bereitgestellt werden, und zum Senden an CloudWatch Container Insights finden Sie in der Dokumentation Container Insights EKS Fargate in der AWS Distribution. OpenTelemetry

Protokolle an Logs senden CloudWatch

Um Protokolle von Ihren Containern an Amazon CloudWatch Logs zu senden, können Sie Fluent Bit oder Fluentd verwenden. Weitere Informationen finden Sie unter Fluent Bit und Fluentd.

Wenn Sie Fluentd nicht bereits verwenden, empfehlen wir die Verwendung von Fluent Bit aus folgenden Gründen:

- Fluent Bit hat einen geringeren Ressourcenbedarf und ist ressourcenschonender in Sachen Speicher- und CPU-Auslastung als Fluentd. Für einen detaillierteren Vergleich siehe Fluent Bit und Fluentd Leistungsvergleich.
- Das Fluent Bit-Image wird von entwickelt und verwaltet. AWS Auf diese AWS Weise können neue Fluent Bit-Image-Funktionen eingeführt und Probleme viel schneller behoben werden.

Themen

Fluent Bit und Fluentd Leistungsvergleich

Die folgenden Tabellen zeigen den Leistungsvorteil, den Fluent Bit gegenüber Fluentd bei Speicher- und CPU-Auslastung hat. Die folgenden Zahlen dienen nur als Referenz und können sich je nach Umgebung ändern.

Protokolle pro Sekunde	Fluentd-CPU-Auslastung	Fluent-Bit-CPU-Auslastung mit Fluentd-kompatibler Konfiguration	Fluent-Bit-CPU-Auslastung mit optimierter Konfiguration
100	0,35 vCPU	0,02 vCPU	0,02 vCPU
1.000	0,32 vCPU	0,14 vCPU	0,11 vCPU
5,000	0,85 vCPU	0,48 vCPU	0,30 vCPU
10.000	0,94 vCPU	0,60 vCPU	0,39 vCPU

Protokolle pro Sekunde	Fluentd-Speicherauslastung	Fluent-Bit-Speichernutzung mit Fluentd-kompatibler Konfiguration	Fluent-Bit-Speichernutzung mit optimierter Konfiguration
100	153 MB	46 MB	37 MB
1.000	270 MB	45 MB	40 MB
5,000	320 MB	55 MB	45 MB
10.000	375 MB	92 MB	75 MB

Richten Sie Fluent Bit ein, um Protokolle an Logs DaemonSet zu senden CloudWatch

Die folgenden Abschnitte helfen Ihnen bei der Bereitstellung von Fluent Bit, um Protokolle von Containern an Logs zu CloudWatch senden.

Themen

- Fehlerbehebung
- Dashboard

Unterschiede, wenn Sie Fluentd bereits verwenden

Wenn Sie Fluentd bereits verwenden, um Protokolle von Containern an Logs zu CloudWatch senden, lesen Sie diesen Abschnitt, um die Unterschiede zwischen Fluentd und Fluent Bit zu erfahren. Wenn Sie Fluentd noch nicht mit Container Insights verwenden, können Sie zu Einrichten von Fluent Bit übergehen.

Wir bieten zwei Standardkonfigurationen für Fluent Bit:

- Fluent Bit-optimierte Konfiguration – Eine Konfiguration, die auf die bewährten Methoden von Fluent Bit ausgerichtet ist.
- Fluentd-kompatible Konfiguration – Eine Konfiguration, die so weit wie möglich am Fluentd-Verhalten ausgerichtet ist.

Die folgende Liste erklärt die Unterschiede zwischen Fluentd und jeder Fluent-Bit-Konfiguration im Detail.

- Unterschiede in den Namen des Protokoll-Streams – Wenn Sie die für Fluent Bit optimierte Konfiguration verwenden, unterscheiden sich die Namen der Protokollstreams.

  Unter `/aws/containerinsights/Cluster_Name/application`

  - Fluent Bit optimierte Konfiguration sendet Protokolle an *`kubernetes-nodeName`*`-application.var.log.containers.`*`kubernetes-podName_kubernetes-namespace_kubernetes-container-name-kubernetes-containerID`*
  - Fluentd sendet Protokolle an *`kubernetes-podName_kubernetes-namespace_kubernetes-containerName_kubernetes-containerID`*

  Unter `/aws/containerinsights/Cluster_Name/host`

  - Fluent Bit optimierte Konfiguration sendet Protokolle an *`kubernetes-nodeName`*`.`*`host-log-file`*
  - Fluentd sendet Protokolle an *`host-log-file-Kubernetes-NodePrivateIp`*

  Unter `/aws/containerinsights/Cluster_Name/dataplane`

  - Fluent Bit optimierte Konfiguration sendet Protokolle an *kubernetes-nodeName*.*dataplaneServiceLog*
  - Fluentd sendet Protokolle an *dataplaneServiceLog*-*Kubernetes-nodeName*
- Die `kube-proxy`- und `aws-node`-Protokolldateien, die Container Insights schreibt, befinden sich an verschiedenen Speicherorten. In der Fluentd-Konfiguration befinden sie sich in `/aws/containerinsights/Cluster_Name/application`. In der für Fluent Bit optimierten Konfiguration befinden sie sich in `/aws/containerinsights/Cluster_Name/dataplane`.
- Die meisten Metadaten wie `pod_name` und `namespace_name` sind in Fluent Bit und Fluentd gleich, aber die folgenden sind unterschiedlich.
  - Die für Fluent Bit optimierte Konfiguration verwendet `docker_id` und Fluentd verwendet `Docker.container_id`.
  - Beide Fluent-Bit-Konfigurationen verwenden nicht die folgenden Metadaten. Sie sind nur in Fluentd vorhanden: `container_image_id`, `master_url`, `namespace_id` und `namespace_labels`.

Einrichten von Fluent Bit

Um Fluent Bit so einzurichten, dass Protokolle von Ihren Containern erfasst werden, können Sie die Schritte unter Schnellstarteinrichtung für Container Insights auf Amazon EKS und Kubernetes oder die Schritte in diesem Abschnitt befolgen.

Bei beiden Methoden muss die an die Cluster-Knoten angefügte IAM-Rolle über ausreichende Berechtigungen verfügen. Weitere Informationen zu den Berechtigungen, die zum Ausführen eines Amazon-EKS-Clusters erforderlich sind, finden Sie unter Amazon-EKS-IAM-Richtlinien, -Rollen und -Berechtigungen im Amazon-EKS-Benutzerhandbuch.

In den folgenden Schritten richten Sie Fluent Bit als DaemonSet ein, um Logs an Logs zu senden. CloudWatch Wenn Sie diesen Schritt abgeschlossen haben, erstellt Fluent Bit die folgenden Protokollgruppen, sofern diese nicht bereits vorhanden ist.

> **Important**
>
> Wenn Sie FluentD bereits in Container Insights konfiguriert haben und FluentD DaemonSet nicht wie erwartet läuft (dies kann passieren, wenn Sie die `containerd` Runtime verwenden), müssen Sie es vor der Installation von Fluent Bit deinstallieren, um zu verhindern, dass Fluent Bit die FluentD-Fehlerprotokollmeldungen verarbeitet. Andernfalls

müssen Sie Fluentd sofort deinstallieren, nachdem Sie Fluent Bit erfolgreich installiert haben. Die Deinstallation von Fluentd nach der Installation von Fluent Bit gewährleistet die Kontinuität der Protokollierung während dieses Migrationsprozesses. Es wird nur eine von Fluent Bit oder FluentD benötigt, um Protokolle an Logs zu senden. CloudWatch

Protokollgruppenname	Protokollquelle
`/aws/containerinsights/` *`Cluster_Name`* `/application`	Alle Protokolldateien in `/var/log/containers`
`/aws/containerinsights/` *`Cluster_Name`* `/host`	Protokolle aus `/var/log/dmesg` , `/var/log/secure` und `/var/log/messages`
`/aws/containerinsights/` *`Cluster_Name`* `/dataplane`	Die Protokolle in `/var/log/journal` für `kubelet.service` , `kubeproxy.service` und `docker.service` .

Um Fluent Bit zu installieren, um Logs von Containern an Logs zu senden CloudWatch

1. Wenn Sie noch keinen Namespace namens `amazon-cloudwatch` haben, erstellen Sie einen, indem Sie den folgenden Befehl eingeben:

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-monitoring/cloudwatch-namespace.yaml
```

2. Führen Sie den folgenden Befehl aus, um einen ConfigMap Namen `cluster-info` mit dem Clusternamen und der Region zu erstellen, an die Protokolle gesendet werden sollen. Ersetzen Sie *`cluster-name`* und *`cluster-region`* durch den Namen und die Region des -Clusters.

```
ClusterName=cluster-name
RegionName=cluster-region
FluentBitHttpPort='2020'
FluentBitReadFromHead='Off'
[[ ${FluentBitReadFromHead} = 'On' ]] && FluentBitReadFromTail='Off'||
 FluentBitReadFromTail='On'
[[ -z ${FluentBitHttpPort} ]] && FluentBitHttpServer='Off' ||
 FluentBitHttpServer='On'
```

```
kubectl create configmap fluent-bit-cluster-info \
--from-literal=cluster.name=${ClusterName} \
--from-literal=http.server=${FluentBitHttpServer} \
--from-literal=http.port=${FluentBitHttpPort} \
--from-literal=read.head=${FluentBitReadFromHead} \
--from-literal=read.tail=${FluentBitReadFromTail} \
--from-literal=logs.region=${RegionName} -n amazon-cloudwatch
```

In diesem Befehl ist das `FluentBitHttpServer` zum Überwachen von Plug-in-Metriken standardmäßig aktiviert. Um es auszuschalten, ändern Sie die dritte Zeile im Befehl in `FluentBitHttpPort=''` (leere Zeichenfolge) im Befehl.

Außerdem liest Fluent Bit standardmäßig Protokolldateien aus dem Trail und erfasst nach der Bereitstellung nur neue Protokolle. Wenn Sie das Gegenteil wünschen, legen Sie `FluentBitReadFromHead='On'` fest und es werden alle Protokolle im Dateisystem gesammelt.

3. Laden Sie das Fluent-Bit-DaemonSet herunter und stellen Sie es für den Cluster bereit, indem Sie den folgenden Befehl ausführen.

   - Wenn Sie die für Fluent Bit optimierte Konfiguration für Linux-Computer verwenden möchten, führen Sie diesen Befehl aus.

     ```
     kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
     cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
     deployment-mode/daemonset/container-insights-monitoring/fluent-bit/fluent-
     bit.yaml
     ```

   - Wenn Sie die für Fluent Bit optimierte Konfiguration für Windows-Computer verwenden möchten, führen Sie diesen Befehl aus.

     ```
     kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
     cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
     deployment-mode/daemonset/container-insights-monitoring/fluent-bit/fluent-bit-
     windows.yaml
     ```

   - Wenn Sie Linux-Computer verwenden und eine Fluent Bit-Konfiguration wünschen, die Fluentd ähnlicher ist, führen Sie diesen Befehl aus.

     ```
     kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
     cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
     ```

```
deployment-mode/daemonset/container-insights-monitoring/fluent-bit/fluent-bit-
compatible.yaml
```

> **Important**
>
> Die Fluent Bit-Daemonset-Konfiguration legt die Protokollebene standardmäßig auf INFO fest, was zu höheren Kosten für die Protokollaufnahme führen kann. CloudWatch Wenn Sie das Aufnahmevolumen und die Kosten von Protokollen reduzieren möchten, können Sie die Protokollebene auf ERROR ändern.
> Weitere Informationen darüber, wie Sie das Protokollvolumen reduzieren, finden Sie unter (Optional) Reduzieren des Protokoll-Volumes von Fluent Bit

4. Überprüfen Sie die Bereitstellung, indem Sie den folgenden Befehl eingeben. Jeder Knoten sollte über eine Pod mit dem Namen fluent-bit-* verfügen.

```
kubectl get pods -n amazon-cloudwatch
```

Die obigen Schritte erstellen die folgenden Ressourcen im Cluster:

- Ein Servicekonto mit dem Namen `Fluent-Bit` im Namespace `amazon-cloudwatch`. Dieses Service-Konto wird zum Ausführen des Fluent Bit daemonSet verwendet. Weitere Informationen finden Sie unter Managing Service Accounts in der Kubernetes-Dokumentation.
- Eine Cluster-Rolle mit dem Namen `Fluent-Bit-role` im Namespace `amazon-cloudwatch`. Dieser Cluster-Rolle gewährt `get`, `list` und `watch` Berechtigungen für Pod-Protokolle an das Servicekonto `Fluent-Bit`. Weitere Informationen finden Sie unter API Overview in der Kubernetes-Dokumentation.
- Ein im Namespace benannter ConfigMap . `Fluent-Bit-config` `amazon-cloudwatch` Dies ConfigMap enthält die Konfiguration, die von Fluent Bit verwendet werden soll. Weitere Informationen finden Sie unter Configure a Pod to Use a ConfigMap in der Dokumentation zu Kubernetes Tasks.

Wenn Sie Ihre Fluent-Bit–Einrichtung überprüfen möchten, führen Sie diese Schritte aus.

Überprüfen der Fluent-Bit-Einrichtung

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

2. Wählen Sie im Navigationsbereich Protokollgruppen aus.
3. Stellen Sie sicher, dass Sie sich in der Region befinden, in der Sie Fluent Bit bereitgestellt haben.
4. Überprüfen Sie die Liste der Protokollgruppen in der Region. Sie sollten Folgendes sehen:
   - `/aws/containerinsights/Cluster_Name/application`
   - `/aws/containerinsights/Cluster_Name/host`
   - `/aws/containerinsights/Cluster_Name/dataplane`
5. Navigieren Sie zu einer dieser Protokollgruppen und überprüfen Sie die letzte Ereigniszeit für die Protokollstreams Wenn es in letzter Zeit relativ zum Zeitpunkt der Bereitstellung von Fluent Bit ist, wird die Einrichtung überprüft.

   Es kann eine leichte Verzögerung beim Erstellen der `/dataplane`-Protokollgruppe geben. Dies ist normal, da diese Protokollgruppen nur erstellt werden, wenn Fluent Bit beginnt, Protokolle für diese Protokollgruppe zu senden.

Unterstützung für mehrzeilige Protokolle

Informationen zur Verwendung von Fluent Bit mit mehrzeiligen Protokollen finden Sie in den folgenden Abschnitten der Fluent-Bit-Dokumentation:

- Mehrzeiliges Parsing
- Mehrzeilig und Container (v1.8)
- Mehrzeiliger Kern (v1.8)
- Immer mehrzeilig für den Fragment-Eingang verwenden

(Optional) Reduzieren des Protokoll-Volumes von Fluent Bit

Standardmäßig senden wir Fluent Bit-Anwendungsprotokolle und Kubernetes-Metadaten an. CloudWatch Wenn Sie das Datenvolumen reduzieren möchten, an das gesendet wird CloudWatch, können Sie verhindern, dass eine oder beide dieser Datenquellen gesendet werden. CloudWatch

Um Fluent-Bit-Anwendungsprotokolle zu beenden, entfernen Sie den folgenden Abschnitt aus der `Fluent-Bit.yaml`-Datei.

```
[INPUT]
        Name                tail
```

```
        Tag                 application.*
        Path                /var/log/containers/fluent-bit*
        Parser              docker
        DB                  /fluent-bit/state/flb_log.db
        Mem_Buf_Limit       5MB
        Skip_Long_Lines     On
        Refresh_Interval    10
```

Um zu verhindern, dass Kubernetes-Metadaten an Protokollereignisse angehängt werden, an die gesendet werden CloudWatch, fügen Sie dem `application-log.conf` Abschnitt in der Datei die folgenden Filter hinzu. `Fluent-Bit.yaml` Ersetzen Sie <Metadata_1>und ähnliche Felder durch die tatsächlichen Metadaten-Identifikatoren.

```
application-log.conf: |
    [FILTER]
        Name                nest
        Match               application.*
        Operation           lift
        Nested_under        kubernetes
        Add_prefix          Kube.

    [FILTER]
        Name                modify
        Match               application.*
        Remove              Kube.<Metadata_1>
        Remove              Kube.<Metadata_2>
        Remove              Kube.<Metadata_3>

    [FILTER]
        Name                nest
        Match               application.*
        Operation           nest
        Wildcard            Kube.*
        Nested_under        kubernetes
        Remove_prefix       Kube.
```

Fehlerbehebung

Wenn diese Protokollgruppen bei Ansicht der richtigen Region nicht finden, überprüfen Sie die Protokolle für die DaemonSet-Fluent-Bit-Pods auf Fehler.

Führen Sie den folgenden Befehl aus und stellen Sie sicher, dass der Status `Running` lautet.

```
kubectl get pods -n amazon-cloudwatch
```

Wenn die Protokolle Fehler im Zusammenhang mit IAM-Berechtigungen enthalten, überprüfen Sie an die Cluster-Knoten angefügte IAM-Rolle. Weitere Informationen zu den Berechtigungen, die zum Ausführen eines Amazon-EKS-Clusters erforderlich sind, finden Sie unter Amazon-EKS-IAM-Richtlinien, -Rollen und -Berechtigungen im Amazon-EKS-Benutzerhandbuch.

Wenn der Pod-Status `CreateContainerConfigError` lautet, rufen Sie den genauen Fehler ab, indem Sie den folgenden Befehl ausführen.

```
kubectl describe pod pod_name -n amazon-cloudwatch
```

Dashboard

Sie können ein Dashboard erstellen, um Metriken jedes ausgeführten Plug-Ins zu überwachen. Sie können Daten für Eingabe- und Ausgabebytes und für Datensatzverarbeitungsraten sowie Ausgabefehler und Wiederholungs-/Fehlgeschlagene Raten sehen. Um diese Metriken anzuzeigen, müssen Sie den CloudWatch Agenten mit der Prometheus-Metrikerfassung für Amazon EKS- und Kubernetes-Cluster installieren. Weitere Informationen zum Einrichten des Dashboards finden Sie unter Installieren Sie den CloudWatch Agenten mit der Erfassung von Prometheus-Metriken auf Amazon EKS- und Kubernetes-Clustern.

> **Note**
>
> Bevor Sie dieses Dashboard einrichten können, müssen Sie Container Insights für Prometheus-Metriken einrichten. Weitere Informationen finden Sie unter Überwachung von Container Insights Prometheus-Metriken.

So erstellen Sie ein Dashboard für die Fluent-Bit-Prometheus-Metriken

1. Erstellen Sie Umgebungsvariablen, indem Sie die Werte auf der rechten Seite in den folgenden Zeilen entsprechend Ihrer Bereitstellung ersetzen.

   ```
   DASHBOARD_NAME=your_cw_dashboard_name
   REGION_NAME=your_metric_region_such_as_us-west-1
   CLUSTER_NAME=your_kubernetes_cluster_name
   ```

2. Erstellen Sie das Dashboard, indem Sie den folgenden Befehl ausführen.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_cloudwatch_dashboards/fluent-bit/cw_dashboard_fluent_bit.json \
| sed "s/{{YOUR_AWS_REGION}}/${REGION_NAME}/g" \
| sed "s/{{YOUR_CLUSTER_NAME}}/${CLUSTER_NAME}/g" \
| xargs -0 aws cloudwatch put-dashboard --dashboard-name ${DASHBOARD_NAME} --
dashboard-body
```

(Optional) Richten Sie Fluentd so ein, dass Protokolle an Logs gesendet werden DaemonSet CloudWatch

> **Warning**
>
> Die Container Insights-Unterstützung für Fluentd befindet sich jetzt im Wartungsmodus, was bedeutet, dass AWS wir keine weiteren Updates für Fluentd bereitstellen werden und dass wir planen, sie in naher future einzustellen. Darüber hinaus verwendet die aktuelle Fluentd-Konfiguration für Container Insights eine alte Version von Fluentd Image `fluent/fluentd-kubernetes-daemonset:v1.10.3-debian-cloudwatch-1.0`, die nicht über die neuesten Verbesserungs- und Sicherheitspatches verfügt. Das neueste Fluentd-Image, das von der Open-Source-Community unterstützt wird, finden Sie unter. fluentd-kubernetes-daemonset
>
> Wir empfehlen dringend, wann immer möglich zur Verwendung FluentBit mit Container Insights zu migrieren. Die Verwendung FluentBit als Log-Forwarder für Container Insights bietet erhebliche Leistungssteigerungen.
>
> Weitere Informationen finden Sie unter Richten Sie Fluent Bit ein, um Protokolle an Logs DaemonSet zu senden CloudWatch und Unterschiede, wenn Sie Fluentd bereits verwenden.

Um Fluentd so einzurichten, dass Protokolle von Ihren Containern erfasst werden, können Sie die Schritte unter Schnellstarteinrichtung für Container Insights auf Amazon EKS und Kubernetes oder die Schritte in diesem Abschnitt befolgen. In den folgenden Schritten richten Sie Fluentd ein, um Logs an Logs DaemonSet zu senden. CloudWatch Wenn Sie diesen Schritt abgeschlossen haben, erstellt Fluentd die folgenden Protokollgruppen, sofern diese nicht bereits vorhanden ist.

Protokollgruppenname	Protokollquelle
/aws/containerinsights/ *Cluster_Name* /application	Alle Protokolldateien in /var/log/containers
/aws/containerinsights/ *Cluster_Name* /host	Protokolle aus /var/log/dmesg , /var/log/secure und /var/log/messages
/aws/containerinsights/ *Cluster_Name* /dataplane	Die Protokolle in /var/log/journal für kubelet.service , kubeproxy.service und docker.service .

Schritt 1: Erstellen Sie einen Namespace für CloudWatch

Verwenden Sie den folgenden Schritt, um einen angeforderten Kubernetes-Namespace zu erstellen. amazon-cloudwatch CloudWatch Sie können diesen Schritt überspringen, wenn Sie diesen Namespace bereits erstellt haben.

Um einen Namespace zu erstellen für CloudWatch

- Geben Sie den folgenden Befehl ein.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-monitoring/cloudwatch-namespace.yaml
```

Schritt 2: Fluentd installieren

Starten Sie diesen Prozess durch Herunterladen von Fluentd. Wenn Sie diese Schritte abgeschlossen haben, werden bei der Bereitstellung die folgenden Ressourcen auf dem Cluster erstellt:

- Ein Servicekonto mit dem Namen fluentd im Namespace amazon-cloudwatch. Dieses Dienstkonto wird verwendet, um DaemonSet Fluentd auszuführen. Weitere Informationen finden Sie unter Managing Service Accounts in der Kubernetes-Dokumentation.
- Eine Cluster-Rolle mit dem Namen fluentd im Namespace amazon-cloudwatch. Dieser Cluster-Rolle gewährt get, list und watch Berechtigungen für Pod-Protokolle an das

Servicekonto `fluentd`. Weitere Informationen finden Sie unter API Overview in der Kubernetes-Dokumentation.

- Ein im ConfigMap Namespace `fluentd-config` benannter. `amazon-cloudwatch` Dies ConfigMap enthält die Konfiguration, die von Fluentd verwendet werden soll. Weitere Informationen finden Sie unter Configure a Pod to Use a ConfigMap in der Dokumentation zu Kubernetes Tasks.

So installieren Sie Fluentd

1. Erstellen Sie einen ConfigMap Namen `cluster-info` mit dem Clusternamen und der AWS Region, an die die Logs gesendet werden sollen. Führen Sie den folgenden Befehl aus, wodurch die Platzhalter mit den Namen Ihres Clusters und Ihrer Region aktualisiert werden.

   ```
   kubectl create configmap cluster-info \
   --from-literal=cluster.name=cluster_name \
   --from-literal=logs.region=region_name -n amazon-cloudwatch
   ```

2. Laden Sie Fluentd herunter und stellen Sie DaemonSet es auf dem Cluster bereit, indem Sie den folgenden Befehl ausführen. Stellen Sie sicher, dass Sie das Container-Image mit der richtigen Architektur verwenden. Das Beispielmanifest funktioniert nur auf x86-Instances und wird `CrashLoopBackOff` eingeben, wenn Sie ARM-Instances (Advanced RISC Machine) in Ihrem Cluster haben. Das Fluentd DaemonSet verfügt nicht über ein offizielles Docker-Image mit mehreren Architekturen, das es Ihnen ermöglicht, ein Tag für mehrere zugrunde liegende Images zu verwenden und die Containerlaufzeit das richtige Image ziehen zu lassen. Das Fluentd-ARM-Image verwendet ein anderes Tag mit einem `arm64`-Suffix.

   ```
   kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
   container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
   daemonset/container-insights-monitoring/fluentd/fluentd.yaml
   ```

   > **Note**
   >
   > Aufgrund einer kürzlich erfolgten Änderung zur Optimierung der Fluentd-Konfiguration und zur Minimierung der Auswirkungen von Fluentd-API-Anforderungen auf Kubernetes-API-Endpunkte wurde die Option „Watch“ für Kubernetes-Filter standardmäßig deaktiviert. Weitere Informationen finden Sie unter fluent-plugin-kubernetes _metadata_filter.

3. Überprüfen Sie die Bereitstellung, indem Sie den folgenden Befehl ausführen. Jeder Knoten sollte über eine Pod mit dem Namen `fluentd-cloudwatch-*` verfügen.

```
kubectl get pods -n amazon-cloudwatch
```

Schritt 3: Fluentd-Einrichtung überprüfen

Gehen Sie zum Überprüfen Ihrer Fluentd-Einrichtung wie folgt vor.

So überprüfen Sie die Fluentd-Einrichtung für Container Insights

1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/cloudwatch/ CloudWatch .
2. Wählen Sie im Navigationsbereich Protokollgruppen aus. Stellen Sie sicher, dass Sie sich in der Region befinden, in der Sie Fluentd für Ihre Container bereitgestellt haben.

   In der Liste der Protokollgruppen in der Region sollte Folgendes angezeigt werden:

   - `/aws/containerinsights/Cluster_Name/application`
   - `/aws/containerinsights/Cluster_Name/host`
   - `/aws/containerinsights/Cluster_Name/dataplane`

   Wenn Sie diese Protokollgruppen sehen, wurde die Fluentd-Einrichtung überprüft.

Unterstützung für mehrzeilige Protokolle

Am 19. August 2019 haben wir die Unterstützung für mehrzeilige Protokolle für die von Fluentd erfassten Protokolle hinzugefügt.

Standardmäßig ist der Starter für mehrzeilige Protokolleinträge ein beliebiges Zeichen ohne Leerzeichen. Das bedeutet, dass alle Protokollzeilen, die mit einem Zeichen beginnen, das keine Leerzeichen enthält, als neuer mehrzeiliger Protokolleintrag betrachtet werden.

Wenn Ihre eigenen Anwendungsprotokolle einen anderen mehrzeiligen Starter verwenden, können Sie diese unterstützen, indem Sie zwei Änderungen in der `fluentd.yaml`-Datei vornehmen.

Schließen Sie sie zunächst von der standardmäßigen mehrzeiligen Unterstützung aus, indem Sie die Pfadnamen Ihrer Protokolldateien zu einem `exclude_path`-Feld im `containers`-Abschnitt von `fluentd.yaml` hinzufügen. Im Folgenden wird ein Beispiel gezeigt.

```
<source>
  @type tail
  @id in_tail_container_logs
  @label @containers
  path /var/log/containers/*.log
  exclude_path ["full_pathname_of_log_file*", "full_pathname_of_log_file2*"]
```

Fügen Sie als Nächstes einen Block für Ihre Protokolldateien zur `fluentd.yaml`-Datei hinzu. Das folgende Beispiel wird für die Protokolldatei des CloudWatch Agenten verwendet, die einen regulären Ausdruck mit Zeitstempel als mehrzeiligen Starter verwendet. Sie können diesen Block kopieren und zu `fluentd.yaml` hinzufügen. Ändern Sie die angegebenen Zeilen, um den Namen der Anwendungsprotokolldatei und den mehrzeiligen Starter widerzuspiegeln, den Sie verwenden möchten.

```
<source>
  @type tail
  @id in_tail_cwagent_logs
  @label @cwagentlogs
  path /var/log/containers/cloudwatch-agent*
  pos_file /var/log/cloudwatch-agent.log.pos
  tag *
  read_from_head true
 <parse>
    @type json
    time_format %Y-%m-%dT%H:%M:%S.%NZ
 </parse>
</source>
```

```
<label @cwagentlogs>
  <filter **>
    @type kubernetes_metadata
    @id filter_kube_metadata_cwagent
  </filter>

  <filter **>
    @type record_transformer
    @id filter_cwagent_stream_transformer
    <record>
```

```
      stream_name ${tag_parts[3]}
    </record>
 </filter>

  <filter **>
    @type concat
    key log
    multiline_start_regexp /^\d{4}[-/]\d{1,2}[-/]\d{1,2}/
    separator ""
    flush_interval 5
    timeout_label @NORMAL
 </filter>

 <match **>
    @type relabel
    @label @NORMAL
 </match>
</label>
```

(Optional) Reduzieren des Protokoll-Volumes von Fluentd

Standardmäßig senden wir Fluentd-Anwendungsprotokolle und Kubernetes-Metadaten an. CloudWatch Wenn Sie das Datenvolumen reduzieren möchten, an das gesendet wird CloudWatch, können Sie verhindern, dass eine oder beide dieser Datenquellen gesendet werden. CloudWatch

Um Fluentd-Anwendungsprotokolle zu beenden, entfernen Sie den folgenden Abschnitt aus der `fluentd.yaml`-Datei.

```
<source>
  @type tail
  @id in_tail_fluentd_logs
  @label @fluentdlogs
  path /var/log/containers/fluentd*
  pos_file /var/log/fluentd.log.pos
  tag *
  read_from_head true
   <parse>
   @type json
   time_format %Y-%m-%dT%H:%M:%S.%NZ
  </parse>
</source>
```

```
<label @fluentdlogs>
  <filter **>
    @type kubernetes_metadata
    @id filter_kube_metadata_fluentd
  </filter>

  <filter **>
    @type record_transformer
    @id filter_fluentd_stream_transformer
    <record>
      stream_name ${tag_parts[3]}
    </record>
   </filter>

   <match **>
     @type relabel
     @label @NORMAL
   </match>
</label>
```

Um zu verhindern, dass Kubernetes-Metadaten an Protokollereignisse angehängt werden, an die gesendet werden CloudWatch, fügen Sie dem `record_transformer` Abschnitt in der Datei eine Zeile hinzu. `fluentd.yaml` Fügen Sie in der Protokollquelle, in der Sie diese Metadaten entfernen möchten, die folgende Zeile hinzu.

```
remove_keys $.kubernetes.pod_id, $.kubernetes.master_url,
 $.kubernetes.container_image_id, $.kubernetes.namespace_id
```

Zum Beispiel:

```
<filter **>
  @type record_transformer
  @id filter_containers_stream_transformer
  <record>
    stream_name ${tag_parts[3]}
  </record>
  remove_keys $.kubernetes.pod_id, $.kubernetes.master_url,
 $.kubernetes.container_image_id, $.kubernetes.namespace_id
</filter>
```

Fehlerbehebung

Wenn Sie diese Protokollgruppen nicht sehen und Sie in der richtigen Region suchen, überprüfen Sie die Protokolle für die DaemonSet Fluentd-Pods, um nach dem Fehler zu suchen.

Führen Sie den folgenden Befehl aus und stellen Sie sicher, dass der Status `Running` lautet.

```
kubectl get pods -n amazon-cloudwatch
```

Notieren Sie in den Ergebnissen des vorherigen Befehls den Pod-Namen, der mit `fluentd-cloudwatch` beginnt. Verwenden Sie diesen Pod-Namen in dem folgenden Befehl.

```
kubectl logs pod_name -n amazon-cloudwatch
```

Wenn die Protokolle Fehler im Zusammenhang mit IAM-Berechtigungen enthalten, überprüfen Sie an die Cluster-Knoten angefügte IAM-Rolle. Weitere Informationen zu den Berechtigungen, die zum Ausführen eines Amazon-EKS-Clusters erforderlich sind, finden Sie unter Amazon-EKS-IAM-Richtlinien, -Rollen und -Berechtigungen im Amazon-EKS-Benutzerhandbuch.

Wenn der Pod-Status `CreateContainerConfigError` lautet, rufen Sie den genauen Fehler ab, indem Sie den folgenden Befehl ausführen.

```
kubectl describe pod pod_name -n amazon-cloudwatch
```

Wenn der Pod-Status `CrashLoopBackOff` ist, stellen Sie sicher, dass die Architektur des Fluentd-Container-Images mit der des Knotens bei der Installation von Fluentd übereinstimmt. Wenn Ihr Cluster sowohl x86- als auch ARM64-Knoten hat, können Sie eine kubernetes.io/arch Beschriftung verwenden, um die Images auf dem richtigen Knoten zu platzieren. Weitere Informationen finden Sie unter kuberntes.io/arch.

(Optional) Richten Sie die Protokollierungauf der Amazon-EKS-Steuerebene ein.

Wenn Sie Amazon EKS verwenden, können Sie optional die Protokollierung der Amazon EKS-Kontrollebene aktivieren, um Prüf- und Diagnoseprotokolle direkt von der Amazon EKS-Steuerebene in CloudWatch Logs bereitzustellen. Weitere Informationen finden Sie unter Amazon EKS-Steuerungsebenenprotokollierung.

(Optional) App-Mesh-Envoy-Zugriffsprotokolle aktivieren

Sie können Container Insights Fluentd so einrichten, dass App Mesh Envoy-Zugriffsprotokolle an Logs gesendet werden. CloudWatch Weitere Informationen finden Sie unter Logging (Protokollierung).

Um Envoy-Zugriffsprotokolle an Logs senden zu lassen CloudWatch

1. Richten Sie Fluentd im Cluster ein. Weitere Informationen finden Sie unter (Optional) Richten Sie Fluentd so ein, dass Protokolle an Logs gesendet werden DaemonSet CloudWatch .
2. Konfigurieren Sie Envoy-Zugriffsprotokolle für Ihre virtuellen Knoten. Detaillierte Anweisungen finden Sie unter Logging (Protokollierung). Stellen Sie sicher, dass Sie den Protokollpfad so konfigurieren, dass sich **`/dev/stdout`** in jedem virtuellen Knoten befindet.

Wenn Sie damit fertig sind, werden die Envoy-Zugriffsprotokolle an die `/aws/containerinsights/`*`Cluster_Name`*`/application` Protokollgruppe gesendet.

(Optional) Das Feature Use_Kubelet für große Cluster aktivieren

Standardmäßig ist die Funktion Use_Kubelet im Kubernetes-Plugin deaktiviert. FluentBit Die Aktivierung dieses Features kann den Datenverkehr zum API-Server reduzieren und ein Engpass-Problem durch den API-Server abmildern. Wir empfehlen, dieses Feature für große Cluster zu aktivieren.

Um Use_Kubelet zu aktivieren, fügen Sie zuerst die Knoten und Knoten-/Proxy-Berechtigungen zur clusterRole-Konfiguration hinzu.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: fluent-bit-role
rules:
  - nonResourceURLs:
      - /metrics
    verbs:
      - get
  - apiGroups: [""]
    resources:
      - namespaces
      - pods
      - pods/logs
```

```
      - nodes
      - nodes/proxy
    verbs: ["get", "list", "watch"]
```

In der DaemonSet Konfiguration benötigt diese Funktion Zugriff auf das Host-Netzwerk. Die Imageversion für `amazon/aws-for-fluent-bit` sollte 2.12.0 oder höher sein bzw. sollte die Fluent Bit-Image-Version 1.7.2 oder höher sein.

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: fluent-bit
  namespace: amazon-cloudwatch
  labels:
    k8s-app: fluent-bit
    version: v1
    kubernetes.io/cluster-service: "true"
spec:
  selector:
    matchLabels:
      k8s-app: fluent-bit
  template:
    metadata:
      labels:
        k8s-app: fluent-bit
        version: v1
        kubernetes.io/cluster-service: "true"
    spec:
      containers:
      - name: fluent-bit
        image: amazon/aws-for-fluent-bit:2.19.0
        imagePullPolicy: Always
        env:
            - name: AWS_REGION
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: logs.region
            - name: CLUSTER_NAME
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: cluster.name
```

```
            - name: HTTP_SERVER
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: http.server
            - name: HTTP_PORT
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: http.port
            - name: READ_FROM_HEAD
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: read.head
            - name: READ_FROM_TAIL
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: read.tail
            - name: HOST_NAME
              valueFrom:
                fieldRef:
                  fieldPath: spec.nodeName
            - name: HOSTNAME
              valueFrom:
                fieldRef:
                  apiVersion: v1
                  fieldPath: metadata.name
            - name: CI_VERSION
              value: "k8s/1.3.8"
          resources:
            limits:
              memory: 200Mi
            requests:
              cpu: 500m
              memory: 100Mi
          volumeMounts:
          # Please don't change below read-only permissions
          - name: fluentbitstate
            mountPath: /var/fluent-bit/state
          - name: varlog
            mountPath: /var/log
            readOnly: true
```

```
        - name: varlibdockercontainers
          mountPath: /var/lib/docker/containers
          readOnly: true
        - name: fluent-bit-config
          mountPath: /fluent-bit/etc/
        - name: runlogjournal
          mountPath: /run/log/journal
          readOnly: true
        - name: dmesg
          mountPath: /var/log/dmesg
          readOnly: true
      terminationGracePeriodSeconds: 10
      hostNetwork: true
      dnsPolicy: ClusterFirstWithHostNet
      volumes:
      - name: fluentbitstate
        hostPath:
          path: /var/fluent-bit/state
      - name: varlog
        hostPath:
          path: /var/log
      - name: varlibdockercontainers
        hostPath:
          path: /var/lib/docker/containers
      - name: fluent-bit-config
        configMap:
          name: fluent-bit-config
      - name: runlogjournal
        hostPath:
          path: /run/log/journal
      - name: dmesg
        hostPath:
          path: /var/log/dmesg
      serviceAccountName: fluent-bit
      tolerations:
      - key: node-role.kubernetes.io/master
        operator: Exists
        effect: NoSchedule
      - operator: "Exists"
        effect: "NoExecute"
      - operator: "Exists"
        effect: "NoSchedule"
```

Die Konfiguration des Kubernetes-Plug-Ins sollte in etwa folgendermaßen aussehen:

```
[FILTER]
        Name                kubernetes
        Match               application.*
        Kube_URL            https://kubernetes.default.svc:443
        Kube_Tag_Prefix     application.var.log.containers.
        Merge_Log           On
        Merge_Log_Key       log_processed
        K8S-Logging.Parser  On
        K8S-Logging.Exclude Off
        Labels              Off
        Annotations         Off
        Use_Kubelet         On
        Kubelet_Port        10250
        Buffer_Size         0
```

Aktualisieren oder Löschen von Container-Insights auf Amazon EKS und Kubernetes

Verwenden Sie die Schritte in diesen Abschnitten, um Ihr CloudWatch Agenten-Container-Image zu aktualisieren oder Container Insights aus einem Amazon EKS- oder Kubernetes-Cluster zu entfernen.

Themen

- Upgrade auf Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS
- Aktualisierung des CloudWatch Agenten-Container-Images
- Den CloudWatch Agenten und Fluent Bit für Container Insights löschen

Upgrade auf Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS

> **Important**
>
> Wenn Sie Container Insights auf einem Amazon EKS-Cluster aktualisieren oder installieren, empfehlen wir Ihnen, das Amazon CloudWatch Observability EKS-Add-on für die Installation zu verwenden, anstatt die Anweisungen in diesem Abschnitt zu verwenden. Um Metriken zur beschleunigten Datenverarbeitung abzurufen, müssen Sie außerdem das Amazon CloudWatch Observability EKS-Add-on verwenden. Weitere Informationen und Anweisungen finden Sie unter Installieren Sie das Amazon CloudWatch Observability EKS-Add-on.

Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS ist die neueste Version von Container Insights. Sie sammelt detaillierte Metriken von Clustern, auf denen Amazon EKS ausgeführt wird, und bietet kuratierte, sofort verwendbare Dashboards, um die Anwendungs- und Infrastrukturtelemetrie detailliert zu untersuchen. Weitere Informationen zu dieser Version von Container Insights finden Sie unter Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS.

Wenn Sie die Originalversion von Container Insights in einem Amazon-EKS-Cluster installiert haben und diese auf die neuere Version mit verbesserter Beobachtbarkeit aktualisieren möchten, folgen Sie den Anweisungen in diesem Abschnitt.

> **Important**
>
> Bevor Sie die Schritte in diesem Abschnitt ausführen können, müssen Sie die Voraussetzungen einschließlich Cert-Manager überprüft haben. Weitere Informationen finden Sie unter Schnellstart mit dem CloudWatch Agenten, Operator und Fluent Bit.

So aktualisieren Sie einen Amazon-EKS-Cluster auf Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS

1. Installieren Sie den CloudWatch Agent-Operator, indem Sie den folgenden Befehl eingeben. *my-cluster-name*Ersetzen Sie es durch den Namen Ihres Amazon EKS- oder Kubernetes-Clusters und *my-cluster-region*ersetzen Sie es durch den Namen der Region, in der die Protokolle veröffentlicht werden. Wir empfehlen, dass Sie dieselbe Region verwenden, in der Ihr Cluster bereitgestellt wird, um die Kosten für AWS ausgehende Datenübertragungen zu reduzieren.

   ```
   ClusterName=my-cluster-name
   RegionName=my-cluster-region
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
   container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
   {{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
    apply -f -
   ```

   Wenn Sie einen Fehler feststellen, der durch widersprüchliche Ressourcen verursacht wird, liegt das wahrscheinlich daran, dass Sie den CloudWatch Agenten und Fluent Bit mit den zugehörigen Komponenten wie dem ServiceAccount, dem ClusterRole und dem bereits auf dem Cluster ClusterRoleBinding installiert haben. Wenn der CloudWatch Agent-Operator versucht,

den CloudWatch Agenten und die zugehörigen Komponenten zu installieren und eine Änderung der Inhalte feststellt, schlägt er die Installation oder Aktualisierung standardmäßig fehl, um zu verhindern, dass der Status der Ressourcen auf dem Cluster überschrieben wird. Es wird empfohlen, alle vorhandenen CloudWatch Agenten mit Container Insights-Setup zu löschen, die Sie zuvor auf dem Cluster installiert hatten, und dann den CloudWatch Agent-Operator zu installieren.

2. (Optional) Um eine bestehende benutzerdefinierte Fluent Bit-Konfiguration anzuwenden, müssen Sie die Configmap aktualisieren, die dem Fluent Bit-Daemonset zugeordnet ist. Der CloudWatch Agent-Operator stellt eine Standardkonfiguration für Fluent Bit bereit, und Sie können die Standardkonfiguration nach Bedarf überschreiben oder ändern. Gehen Sie wie folgt vor, um eine benutzerdefinierte Konfiguration anzuwenden.

   a. Öffnen Sie die bestehende Konfiguration, indem Sie den folgenden Befehl eingeben.

   ```
   kubectl edit cm fluent-bit-config -n amazon-cloudwatch
   ```

   b. Nehmen Sie Ihre Änderungen in der Datei vor und geben Sie dann die Eingabetaste ein, `:wq` um die Datei zu speichern und den Bearbeitungsmodus zu beenden.

   c. Starten Sie Fluent Bit neu, indem Sie den folgenden Befehl eingeben.

   ```
   kubectl rollout restart fluent-bit -n amazon-cloudwatch
   ```

Aktualisierung des CloudWatch Agenten-Container-Images

> **Important**
>
> Wenn Sie Container Insights auf einem Amazon EKS-Cluster aktualisieren oder installieren, empfehlen wir Ihnen, das Amazon CloudWatch Observability EKS-Add-on für die Installation zu verwenden, anstatt die Anweisungen in diesem Abschnitt zu verwenden. Um Metriken zur beschleunigten Datenverarbeitung abzurufen, müssen Sie außerdem das Amazon CloudWatch Observability EKS-Add-on oder den CloudWatch Agent-Operator verwenden. Weitere Informationen und Anweisungen finden Sie unter Installieren Sie das Amazon CloudWatch Observability EKS-Add-on.

Wenn Sie Ihr Container-Image auf die neueste Version aktualisieren müssen, führen Sie die Schritte in diesem Abschnitt aus.

So aktualisieren Sie Ihr Container-Image

1. Überprüfen Sie, ob die `amazoncloudwatchagent` Customer Resource Definition (CRD) bereits existiert, indem Sie den folgenden Befehl eingeben.

   ```
   kubectl get crds amazoncloudwatchagents.cloudwatch.aws.amazon.com -n amazon-
   cloudwatch
   ```

   Wenn dieser Befehl die Fehlermeldung zurückgibt, dass die CRD fehlt, verfügt der Cluster nicht über Container Insights with enhanced observabilit for Amazon EKS, die CloudWatch mit dem Agent-Operator konfiguriert wurde. Lesen Sie in diesem Fall Upgrade auf Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS.
2. Wenden Sie die neueste `cwagent-version.yaml`-Datei an, indem Sie den folgenden Befehl eingeben.

   ```
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
   insights/main/k8s-quickstart/cwagent-version.yaml | kubectl apply -f -
   ```

Den CloudWatch Agenten und Fluent Bit für Container Insights löschen

Wenn Sie Container Insights mithilfe der Installation des CloudWatch Observability-Add-ons für Amazon EKS installiert haben, können Sie Container Insights und den CloudWatch Agenten löschen, indem Sie den folgenden Befehl eingeben:

> **Note**
>
> Das Amazon EKS-Add-on unterstützt jetzt Container Insights auf Windows-Worker-Knoten. Wenn Sie das Amazon EKS-Add-on löschen, wird Container Insights for Windows ebenfalls gelöscht.

```
aws eks delete-addon –cluster-name my-cluster –addon-name amazon-cloudwatch-
observability
```

Andernfalls geben Sie den folgenden Befehl ein, um alle Ressourcen zu löschen, die sich auf den CloudWatch Agenten und Fluent Bit beziehen. In diesem Befehl ist *My_Cluster_Name der Name*

Ihres Amazon EKS- oder Kubernetes-Clusters und *My_Region der Name der Region*, in der die Protokolle veröffentlicht werden.

```
ClusterName=My_Cluster_Name
RegionName=My-Region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 delete -f -
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/main/k8s-quickstart/cwagent-custom-resource-definitions.yaml | kubectl delete
 -f -
```

Anzeigen von Container-Insights-Metriken

Nachdem Sie Container Insights eingerichtet haben und es Metriken sammelt, können Sie diese Metriken in der Konsole anzeigen. CloudWatch

Damit Container Insights-Metriken auf Ihrem Dashboard angezeigt werden, müssen Sie die Container Insights-Einrichtung abschließen. Weitere Informationen finden Sie unter Einrichten von Container Insights.

Dieses Verfahren erläutert, wie Sie die Metriken anzeigen, die Container Insights automatisch aus den gesammelten Protokolldaten generiert. Im Rest dieses Abschnitts wird erklärt, wie Sie Ihre Daten weiter untersuchen und CloudWatch Logs Insights verwenden können, um mehr Metriken mit größerer Granularität zu erhalten.

So zeigen Sie Container Insights-Metriken an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Instances und anschließend Container Insights aus.
3. Wählen Sie im Dropdown-Feld unter Container Insights die Option Leistungsüberwachung aus.
4. Verwenden Sie die Dropdown-Felder oben, um den Typ der Ressource, die angezeigt werden soll, sowie die spezifische Ressource auszuwählen.

Sie können einen CloudWatch Alarm für jede Metrik einrichten, die Container Insights erfasst. Weitere Informationen finden Sie unter CloudWatch Amazon-Alarme verwenden.

> **Note**
>
> Wenn Sie CloudWatch Application Insights bereits für die Überwachung Ihrer containerisierten Anwendungen eingerichtet haben, wird das Application Insights-Dashboard unter dem Container Insights-Dashboard angezeigt. Wenn Sie Application Insights noch nicht aktiviert haben, können Sie dies tun, indem Sie unter der Leistungsanzeige im Container-Insights-Dashboard Auto-configure Application Insights (Application Insights automatisch konfigurieren) auswählen.
> Weitere Informationen zu Application Insights und containerisierten Anwendungen finden Sie unter Aktivieren von Application Insights zur Ressourcenüberwachung für Amazon ECS und Amazon EKS.

Anzeigen von Hauptbeitragenden

Für einige Ansichten in der Leistungsüberwachung von Container Insights können Sie auch die wichtigsten Mitwirkenden nach Arbeitsspeicher oder CPU oder den zuletzt aktiven Ressourcen anzeigen. Dies ist verfügbar, wenn Sie eines der folgenden Dashboards im Dropdown-Feld oben auf der Seite auswählen:

- ECS-Services
- ECS-Aufgaben
- EKS-Namespaces
- EKS-Services
- EKS-Pods

Wenn Sie einen dieser Arten von Ressourcen anzeigen, wird am unteren Rand der Seite eine Tabelle angezeigt, die zunächst nach CPU-Auslastung sortiert ist. Sie können es ändern, um nach Speicherauslastung oder letzter Aktivität zu sortieren. Um mehr über eine der Zeilen in der Tabelle anzuzeigen, können Sie das Kontrollkästchen neben dieser Zeile aktivieren und dann Aktionen auswählen und eine der Optionen im Menü Aktionen auswählen.

Verwenden von CloudWatch Logs Insights zum Anzeigen von Container Insights-Daten

Container Insights sammelt Metriken mithilfe von Leistungsprotokollereignissen mit eingebettetem Metrikformat. Die Protokolle werden in CloudWatch Logs gespeichert. CloudWatch generiert

automatisch mehrere Metriken aus den Protokollen, die Sie in der CloudWatch Konsole einsehen können. Sie können auch eine eingehendere Analyse der Leistungsdaten durchführen, die mithilfe von CloudWatch Logs Insights-Abfragen gesammelt werden.

Weitere Informationen zu CloudWatch Logs Insights finden Sie unter Analysieren von Protokolldaten mit CloudWatch Logs Insights. Weitere Informationen über die Protokollfelder, die Sie in Abfragen verwenden können, finden Sie unter Container Insights-Performance-Protokollereignisse für Amazon EKS und Kubernetes.

So verwenden Sie CloudWatch Logs Insights, um Ihre Container-Metrikdaten abzufragen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Insights aus.

   Oben auf dem Bildschirm befindet sich der Abfrageeditor. Wenn Sie CloudWatch Logs Insights zum ersten Mal öffnen, enthält dieses Feld eine Standardabfrage, die die 20 neuesten Protokollereignisse zurückgibt.
3. Wählen Sie im Feld über dem Abfrageeditor eine der Container Insights-Protokollgruppen aus, die abgefragt werden soll. Damit die folgenden Abfragen ordnungsgemäß ausgeführt werden, muss der Name der Protokollgruppe mit performance enden.

   Wenn Sie eine Protokollgruppe auswählen, erkennt CloudWatch Logs Insights automatisch Felder in den Daten in der Protokollgruppe und zeigt sie im rechten Bereich unter Entdeckte Felder an. Dort finden Sie auch ein Balkendiagramm der Protokollereignisse in dieser Protokollgruppe im Zeitverlauf. Dieses Balkendiagramm zeigt die Verteilung der Ereignisse in der Protokollgruppe, die Ihrer Abfrage und Ihrem Zeitraum entspricht, nicht nur die in der Tabelle angezeigten Ereignisse.
4. Ersetzen Sie im Abfrageeditor die Standard-Abfrage durch die folgende Abfrage und wählen Sie Run query (Abfrage ausführen).

   ```
   STATS avg(node_cpu_utilization) as avg_node_cpu_utilization by NodeName
   | SORT avg_node_cpu_utilization DESC
   ```

   Diese Abfrage zeigt eine Liste von Knoten an, die nach durchschnittlicher Knoten-CPU-Auslastung sortiert ist.
5. Um ein weiteres Beispiel zu testen, ersetzen Sie diese Abfrage durch eine andere Abfrage und wählen Sie Run query (Abfrage ausführen) aus. Weitere Beispielabfragen finden Sie weiter unten auf dieser Seite.

```
STATS avg(number_of_container_restarts) as avg_number_of_container_restarts by
 PodName
| SORT avg_number_of_container_restarts DESC
```

Diese Abfrage zeigt eine Liste Ihrer Pods an, die nach durchschnittlicher Anzahl von Container-Neustarts sortiert ist.

6. Wenn Sie eine andere Abfrage testen möchten, können Sie Einschlussfelder aus der Liste auf der rechten Seite des Bildschirms verwenden. Weitere Informationen zur Abfragesyntax finden Sie unter CloudWatch Logs Insights-Abfragesyntax.

So zeigen Sie Listen Ihrer Ressourcen an:

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Resources aus.
3. Die Standardansicht ist eine Liste Ihrer Ressourcen, die von Container Insights überwacht werden, sowie der Alarme, die Sie für diese Ressourcen eingerichtet haben. Um eine visuelle Karte der Ressourcen anzuzeigen, wählen Sie Map view (Kartenansicht).
4. In der Kartenansicht können Sie mit dem Mauszeiger auf eine beliebige Ressource in der Karte zeigen, um grundlegende Metriken zu dieser Ressource anzuzeigen. Sie können eine beliebige Ressource auswählen, um detailliertere Diagramme dazu anzuzeigen.

Anwendungsfall: Metriken auf Aufgabenebene in Amazon ECS-Containern anzeigen

Das folgende Beispiel zeigt, wie Sie CloudWatch Logs Insights verwenden können, um tiefer in Ihre Container Insights-Logs einzutauchen. Weitere Beispiele finden Sie im Blog Introducing Amazon CloudWatch Container Insights for Amazon ECS.

Container Insights generiert nicht automatisch Metriken auf der Aufgaben-Ebene der Granularität. Die folgende Abfrage zeigt Metriken auf Aufgaben-Ebene für die CPU- und Arbeitsspeichernutzung an.

```
stats avg(CpuUtilized) as CPU, avg(MemoryUtilized) as Mem by TaskId, ContainerName
| sort Mem, CPU desc
```

Andere Beispielabfragen für Container Insights

Liste Ihrer Pods, sortiert nach durchschnittlicher Anzahl der Container-Neustarts

```
STATS avg(number_of_container_restarts) as avg_number_of_container_restarts by PodName
| SORT avg_number_of_container_restarts DESC
```

Angefragte Pods im Vergleich zu ausgeführten Pods

```
fields @timestamp, @message
| sort @timestamp desc
| filter Type="Pod"
| stats min(pod_number_of_containers) as requested,
 min(pod_number_of_running_containers) as running, ceil(avg(pod_number_of_containers-
pod_number_of_running_containers)) as pods_missing by kubernetes.pod_name
| sort pods_missing desc
```

Anzahl der Cluster-Knotenausfälle

```
stats avg(cluster_failed_node_count) as CountOfNodeFailures
| filter Type="Cluster"
| sort @timestamp desc
```

Anwendungsprotokollfehler nach Container-Name

```
stats count() as countoferrors by kubernetes.container_name
| filter stream="stderr"
| sort countoferrors desc
```

Von Container Insights erfasste Metriken

Container Insights erfasst einen Satz von Metriken für Amazon ECS und AWS Fargate Amazon ECS und einen anderen Satz für Amazon EKS, AWS Fargate Amazon EKS und Kubernetes.

Die Metriken sind erst sichtbar, wenn die Container-Aufgaben bereits einige Zeit laufen.

Themen

- Amazon-ECS-Container-Insights-Metriken
- Container-Insights-Metriken für Amazon EKS und Kubernetes

Amazon-ECS-Container-Insights-Metriken

Die folgende Tabelle listet die Metriken und Dimensionen auf, die von Container Insights für Amazon ECS erfasst werden. Diese Metriken befinden sich im `ECS/ContainerInsights`-Namespace. Weitere Informationen finden Sie unter Metriken.

Wenn Sie keine Container Insights-Metriken in Ihrer Konsole sehen, stellen Sie sicher, dass Sie die Einrichtung von Container Insights durchgeführt haben. Metriken werden erst angezeigt, wenn Container Insights vollständig eingerichtet wurde. Weitere Informationen finden Sie unter Einrichten von Container Insights.

Die folgenden Metriken sind verfügbar, wenn Sie die Schritte in Einrichten von Container Insights in Amazon ECS für Cluster- und Service-Level-Metriken ausführen

Metrikname	Dimensionen	Beschreibung
`ContainerInstanceCount`	ClusterName	Die Anzahl der EC2-Instances, die den Amazon-ECS-Agenten ausführen und bei einem Cluster registriert sind. Diese Metrik wird nur für Container-Instances erfasst, die Amazon-ECS-Aufgaben im Cluster ausführen. Sie wird nicht für leere Container-Instances erfasst, die keine Amazon-ECS-Aufgaben haben. Einheit: Anzahl

`CpuUtilized`	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
		Die CPU-Einheiten, die von Aufgaben in der Ressource genutzt werden, die durch den von Ihnen verwendet

Metrikname	Dimensionen	Beschreibung
	`ClusterName`	en Dimensionssatz angegeben wird. Diese Metrik wird nur für Aufgaben erfasst, die über eine definiert e CPU-Reservierung in ihrer Aufgabendefinition verfügen. Einheit: keine

`CpuReserved`	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		Die CPU-Einheiten, die von Aufgaben in der Ressource reserviert werden, die durch den von Ihnen verwendet en Dimensionssatz angegeben wird.
		
		Diese Metrik wird nur für Aufgaben erfasst, die über eine definiert e CPU-Reservierung in ihrer Aufgabendefinition verfügen.
		
		Einheit: keine

`DeploymentCount`	`ServiceName` , `ClusterName`	Die Anzahl der Bereitste llungen in einem Amazon-ECS-Service. Einheit: Anzahl

Metrikname	Dimensionen	Beschreibung
`DesiredTaskCount`	`ServiceName` , `ClusterName`	Die gewünschte Anzahl von Aufgaben für einen Amazon-ECS-Service. Einheit: Anzahl

`EBSFilesystemSize`	`VolumeName` , `TaskDefinitionFamily` , `ClusterName`
	
	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
		Die Gesamtmenge des Amazon EBS-Dateisystemspeichers in Gigabyte (GB), der den Ressourcen zugewiesen ist, die durch die von Ihnen verwendeten Dimensionen spezifiziert sind.
		
		Diese Metrik ist nur für Aufgaben verfügbar, die auf der Amazon ECS-Infrastruktur ausgeführt werden, die auf Fargate mit Plattformversion `1.4.0` oder Amazon EC2 EC2-Instances mit der Container-Agent-Version `1.79.0` oder höher ausgeführt wird.
		
		Einheit: Gigabyte (GB)

Metrikname	Dimensionen	Beschreibung
EBSFilesystemUtilized	VolumeName , TaskDefinitionFamily , ClusterName TaskDefinitionFamily , ClusterName ServiceName , ClusterName	Die Gesamtmenge in Gigabyte (GB) des Amazon EBS-Dateisystemspeichers, der von den Ressourcen verwendet wird, die in den von Ihnen verwendeten Dimensionen spezifiziert sind. Diese Metrik ist nur für Aufgaben verfügbar, die auf der Amazon ECS-Infrastruktur ausgeführt werden, die auf Fargate mit Plattformversion 1.4.0 oder Amazon EC2 EC2-Instances mit der Container-Agent-Version 1.79.0 oder höher ausgeführt wird. Für Aufgaben, die auf Fargate ausgeführt werden, reserviert Fargate Speicherplatz auf der Festplatte, der nur von Fargate verwendet wird. Für den Speicherplatz, den Fargate verwendet, fallen keine Kosten an, aber Sie werden diesen zusätzlichen Speicherplatz mithilfe von Tools wie df sehen.

Metrikname	Dimensionen	Beschreibung
		Einheit: Gigabyte (GB)

`EphemeralStorageReserved` [1]	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		Die Anzahl der Bytes, die aus dem Speicher in der Ressource reserviert wurden, die durch die von Ihnen verwendeten Dimensionen angegeben wird. Ephemerer Speicher wird für das Root-Dateisystem des Containers und alle im Container-Image und der Aufgabendefinition definierten Bind-Mount-Host-Volumes verwendet. Die Menge des kurzlebigen Speichers kann in einer laufenden Aufgabe nicht geändert werden.
		
		Diese Metrik ist nur für Aufgaben verfügbar, die auf Linux-Plattformversion 1.4.0 oder höher ausgeführt werden.
		
		Einheit: Gigabyte (GB)

Metrikname	Dimensionen	Beschreibung

`EphemeralStorageUtilized` [1]	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		Die Anzahl der Bytes, die vom ephemeren Speicher in der Ressource verwendet werden, die durch die von Ihnen verwendeten Dimension en angegeben ist. Ephemerer Speicher wird für das Root-Dateisystem des Containers und alle im Container-Image und der Aufgabendefinition definierten Bind-Mount-Host-Volumes verwendet . Die Menge des kurzlebig en Speichers kann in einer laufenden Aufgabe nicht geändert werden.
		
		Diese Metrik ist nur für Aufgaben verfügbar, die auf Linux-Plattformver sion 1.4.0 oder höher ausgeführt werden.
		
		Einheit: Gigabyte (GB)

Metrikname	Dimensionen	Beschreibung

`MemoryUtilized`	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		Der Arbeitsspeicher, der von Aufgaben in der Ressource genutzt wird, die durch den von Ihnen verwendeten Dimensionssatz angegeben wird.
		
		Diese Metrik wird nur für Aufgaben erfasst, die über eine definierte Speicherreservierung in ihrer Aufgabendefinition verfügen.
		
		Einheit: Megabyte

`MemoryReserved`	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		Der Arbeitsspeicher, der von Aufgaben in der Ressource reserviert wird, die durch den von Ihnen verwendeten Dimensionssatz angegeben wird.
		
		Diese Metrik wird nur für Aufgaben erfasst, die über eine definierte Speicherreservierung in ihrer Aufgabendefinition verfügen.
		
		Einheit: Megabyte

Metrikname	Dimensionen	Beschreibung

`NetworkRxBytes`	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		Die Anzahl der Bytes, die von der Ressource empfangen wurden, die durch die von Ihnen verwendeten Dimensionen angegeben wird. Diese Metrik wird aus der Docker-Laufzeit abgerufen.
		
		Diese Metrik ist nur für Container in Aufgaben verfügbar, die den `awsvpc`- oder `bridge`-Netzwerkmodus verwenden.
		
		Einheit: Byte/Sekunde

Metrikname	Dimensionen	Beschreibung
NetworkTxBytes	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	Die Anzahl der Bytes, die von der Ressource übertragen wurden, die durch die von Ihnen verwendeten Dimension en angegeben wird. Diese Metrik wird aus der Docker-Laufzeit abgerufen. Diese Metrik ist nur für Container in Aufgaben verfügbar, die den awsvpc- oder bridge-Netzwerkmodus verwenden. Einheit: Byte/Sekunde
PendingTaskCount	ServiceName , ClusterName	Die Anzahl der Aufgaben, die sich momentan im Status PENDING befinden. Einheit: Anzahl
RunningTaskCount	ServiceName , ClusterName	Die Anzahl der Aufgaben, die sich momentan im Status RUNNING befinden. Einheit: Anzahl
ServiceCount	ClusterName	Die Anzahl der Services im Cluster. Einheit: Anzahl

Metrikname	Dimensionen	Beschreibung
StorageReadBytes	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	Die Anzahl der Bytes, die aus dem Speicher auf der Instance in der Ressource gelesen wurden, die durch die von Ihnen verwendeten Dimensionen angegeben wird. Dies beinhalte t nicht die gelesenen Bytes für Ihre Speicherg eräte. Diese Metrik wird aus der Docker-Laufzeit abgerufen. Einheit: Byte
StorageWriteBytes	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	Die Anzahl der Bytes, die in den Speicher in der Ressource geschrieben wurden, die durch die von Ihnen verwendeten Dimensionen angegeben wird. Diese Metrik wird aus der Docker-Laufzeit abgerufen. Einheit: Byte
TaskCount	ClusterName	Die Anzahl der Aufgaben, die im Cluster ausgeführt werden. Einheit: Anzahl

Metrikname	Dimensionen	Beschreibung
`TaskSetCount`	`ServiceName` , `ClusterName`	Die Anzahl der Aufgabensätze im Service. Einheit: Anzahl

> **Note**
>
> Die `EphemeralStorageReserved`- und `EphemeralStorageUtilized`-Metriken und sind nur für Aufgaben verfügbar, die auf Linux-Plattformversion 1.4.0 oder höher ausgeführt werden.
> Fargate reserviert Speicherplatz auf der Festplatte. Er wird nur von Fargate verwendet. Ihnen entstehen dafür keine Kosten. Es wird in diesen Metriken nicht angezeigt. Sie können diesen zusätzlichen Speicherplatz jedoch in anderen Tools sehen, wie z. B. `df`.

Die folgenden Metriken sind verfügbar, wenn Sie die Schritte in Bereitstellung des CloudWatch Agenten zur Erfassung von Metriken auf EC2-Instanzebene auf Amazon ECS ausführen

Metrikname	Dimensionen	Beschreibung
`instance_cpu_limit`	`ClusterName`	Die maximale Anzahl von CPU-Einheiten, die einer einzelnen EC2-Instance im Cluster zugewiesen werden können. Einheit: keine

`instance_cpu_reserved_capacity`	`ClusterName`
	
	`InstanceId` , `ContainerInstanceId` , `ClusterName`
		Der Prozentsatz der CPU, die derzeit für eine einzelne EC2-Instance im Cluster reserviert wird.

Metrikname	Dimensionen	Beschreibung
		Einheit: Prozent
instance_cpu_usage_total	ClusterName	Die Anzahl der CPU-Einheiten, die auf einer einzelnen EC2-Instance im Cluster verwendet werden. Einheit: keine
instance_cpu_utilization	ClusterName InstanceId , ContainerInstanceId , ClusterName	Der Gesamtprozentsatz der CPU-Einheiten, die auf einer einzelnen EC2-Instance im Cluster verwendet werden. Einheit: Prozent
instance_filesystem_utilization	ClusterName InstanceId , ContainerInstanceId , ClusterName	Der Gesamtprozentsatz der Dateisystemkapazität, die auf einer einzelnen EC2-Instance im Cluster verwendet wird. Einheit: Prozent
instance_memory_limit	ClusterName	Die maximale Menge an Arbeitsspeicher in Byte, die einer einzelnen EC2-Instance in diesem Cluster zugewiesen werden kann. Einheit: Byte

Metrikname	Dimensionen	Beschreibung
instance_memory_reserved_capacity	ClusterName InstanceId , ContainerInstanceId , ClusterName	Der Prozentsatz des Arbeitsspeichers, der derzeit für eine einzelne EC2-Instance im Cluster reserviert wird. Einheit: Prozent
instance_memory_utilization	ClusterName InstanceId , ContainerInstanceId , ClusterName	Der Gesamtprozentsatz des Arbeitsspeichers, der auf einer einzelnen EC2-Instance im Cluster verwendet wird. Einheit: Prozent
instance_memory_working_set	ClusterName	Die Menge an Arbeitsspeicher (in Byte), die auf einer einzelnen EC2-Instance im Cluster verwendet wird. Einheit: Byte

Metrikname	Dimensionen	Beschreibung
`instance_network_total_bytes`	`ClusterName`	Die Gesamtanzahl der Bytes pro Sekunde, die über das Netzwerk auf einer einzelnen EC2-Instance im Cluster übertragen und empfangen werden. Einheit: Byte/Sekunde
`instance_number_of_running_tasks`	`ClusterName`	Die Anzahl der ausgeführten Aufgaben auf einer einzelnen EC2-Instance im Cluster. Einheit: Anzahl

Container-Insights-Metriken für Amazon EKS und Kubernetes

In den folgenden Tabellen sind die Metriken und Dimensionen aufgeführt, die Container Insights für Amazon EKS und Kubernetes erfasst. Diese Metriken befinden sich im `ContainerInsights`-Namespace. Weitere Informationen finden Sie unter Metriken.

Wenn Sie keine Container Insights-Metriken in Ihrer Konsole sehen, stellen Sie sicher, dass Sie die Einrichtung von Container Insights durchgeführt haben. Metriken werden erst angezeigt, wenn Container Insights vollständig eingerichtet wurde. Weitere Informationen finden Sie unter Einrichten von Container Insights.

Wenn Sie Version 1.5.0 oder höher des Amazon EKS-Add-ons oder Version 1.300035.0 des CloudWatch Agenten verwenden, werden die meisten in der folgenden Tabelle aufgeführten Metriken sowohl für Linux- als auch für Windows-Knoten erfasst. In der Spalte Metrikname der Tabelle können Sie sehen, welche Metriken für Windows nicht erfasst werden.

In der Originalversion von Container Insights werden die Metriken als benutzerdefinierte Metriken berechnet. Bei Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS werden die Container-Insights-Metriken pro Beobachtung abgerechnet, anstatt pro gespeicherter Metrik oder aufgenommenem Protokoll. Weitere Informationen zur CloudWatch Preisgestaltung finden Sie unter CloudWatch Amazon-Preise.

> **Note**
>
> Unter Windows `pod_network_tx_bytes` werden Netzwerkmetriken wie `pod_network_rx_bytes` und nicht für Host-Prozesscontainer erfasst.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
`cluster_failed_node_count`	`ClusterName`		Die Anzahl der fehlgeschlagenen Workerknoten im Cluster. Ein Knoten gilt als ausgefallen, wenn er unter Knotenbedingungen leidet. Weitere Informationen finden Sie unter Bedingungen in der Kubernetes-Dokumentation.
`cluster_node_count`	`ClusterName`		Die Gesamtzahl der Workerknoten im Cluster.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`namespace_number_of_running_pods`	`Namespace` `ClusterName`
	
	`ClusterName`
			Die Anzahl der Pods, die pro Namespace in der Ressource ausgeführt werden, die durch die von Ihnen verwendeten Dimensionen angegeben wird.

`node_cpu_limit`	`ClusterName`	`ClusterName` , `InstanceId` , `NodeName`	Die maximale Anzahl der CPU-Einheiten, die einem einzelnen Knoten in diesem Cluster zugewiesen werden können.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`node_cpu_reserved_capacity`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			Der Prozentsatz der CPU-Einheiten, die für Knotenkomponenten, wie z. B. kubelet, kube-proxy und Docker, reserviert sind.
			
			Formel: `node_cpu_request / node_cpu_limit`
			
			Note
			`node_cpu_request` wird nicht direkt als Metrik gemeldet, sondern ist ein Feld in Leistungsprotokoll-Ereignissen. Weitere Informationen finden Sie unter Relevante

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			Felder in Performance-Protokollereignissen für Amazon EKS und Kubernetes.
`node_cpu_usage_total`	`ClusterName`	`ClusterName` , `InstanceId` , `NodeName`	Die Anzahl der CPU-Einheiten, die auf den Knoten im Cluster verwendet werden.

`node_cpu_utilization`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			Der Gesamtprozentsatz der CPU-Einheiten, die auf den Knoten im Cluster verwendet werden.
			
			Formel: `node_cpu_usage_total / node_cpu_limit`

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`node_filesystem_utilization`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			Der Gesamtprozentsatz der Dateisystemkapazität, die auf den Knoten im Cluster verwendet wird.
			
			Formel: `node_filesystem_usage / node_filesystem_capacity`
			
			Note
			`node_filesystem_usage` und `node_filesystem_capacity` werden nicht direkt als Metriken gemeldet, sondern sind Felder in Leistungsprotokoll-Ereignissen. Weitere

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			Informationen finden Sie unter Relevante Felder in Performance-Protokollereignissen für Amazon EKS und Kubernetes.
`node_memory_limit`	`ClusterName`	`ClusterName` , `InstanceId` , `NodeName`	Die maximale Menge an Arbeitsspeicher in Byte, die einem einzelnen Knoten in diesem Cluster zugewiesen werden kann.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`node_file system_inodes`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar. Sie ist unter Windows nicht verfügbar.
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Die Gesamtzahl der inodes (verwendet und unbenutzt) auf einem Knoten.

`node_file system_in odes_free`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar. Sie ist unter Windows nicht verfügbar.
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Die Anzahl der ungenutzten inodes auf einem Knoten.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`node_memory_reserved_capacity`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			Der Prozentsatz des Arbeitsspeichers, der derzeit auf den Knoten im Cluster verwendet wird.
			
			Formel: `node_memory_request / node_memory_limit`
			
			Note
			`node_memory_request` wird nicht direkt als Metrik gemeldet, sondern ist ein Feld in Leistungsprotokoll-Ereignissen. Weitere Informationen finden Sie unter Relevante Felder in

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			Performance-Protokollereignissen für Amazon EKS und Kubernetes.
node_memory_utilization	NodeName, ClusterName , InstanceId ClusterName		Der Prozentsatz des Arbeitsspeichers, der derzeit vom Knoten oder den Knoten verwendet wird. Dies ist der Prozentsatz der Knotenspeichernutzung geteilt durch die Knotenspeicherbegrenzung. Formel: node_memory_working_set / node_memory_limit .
node_memory_working_set	ClusterName	ClusterName , InstanceId , NodeName	Die Menge an Arbeitsspeicher in Byte, die im arbeitenden Satz der Knoten im Cluster verwendet wird.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`node_network_total_bytes`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			Die Gesamtzahl der pro Knoten in einem Cluster über das Netzwerk gesendeten und empfangenen Bytes pro Sekunde.
			
			Formel: `node_network_rx_bytes + node_network_tx_bytes`
			
			Note
			`node_network_rx_bytes` und `node_network_tx_bytes` werden nicht direkt als Metriken gemeldet, sondern sind Felder in Leistungsprotokoll-Ereignissen. Weitere Informati

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			onen finden Sie unter Relevante Felder in Performance-Protokollereignissen für Amazon EKS und Kubernetes.

`node_number_of_running_containers`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			Die Anzahl der pro Knoten in einem Cluster ausgeführten Container.

`node_number_of_running_pods`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			Die Anzahl der pro Knoten in einem Cluster ausgeführten Pods.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`node_status_allocatable_pods`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Die Anzahl der Pods, die einem Knoten auf der Grundlage seiner zuweisbaren Ressourcen zugewiesen werden können. Diese ist definiert als die verbleibende Kapazität eines Knotens nach Berücksichtigung der Reservierungen von System-Daemons und der harten Schwellenwerte für die Bereinigung.

`node_status_capacity_pods`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Die Anzahl der Pods, die einem Knoten basierend auf seiner Kapazität zugewiesen werden können.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`node_status_condition_ready`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Zeigt an, ob die Knotenstatusbedingung `Ready` wahr ist.

`node_status_condition_memory_pressure`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Zeigt an, ob die Knotenstatusbedingung `MemoryPressure` wahr ist.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`node_status_condition_pid_pressure`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Zeigt an, ob die Knotenstatusbedingung `PIDPressure` wahr ist.

`node_status_condition_disk_pressure`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Zeigt an, ob die Knotenstatusbedingung `OutOfDisk` wahr ist.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`node_status_condition_unknown`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Gibt an, ob eine der Knotenstatusbedingungen Unbekannt ist.

`node_interface_network_rx_dropped`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Die Anzahl der Pakete, die von einer Netzwerkschnittstelle auf dem Knoten empfangen und anschließend verworfen wurden.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`node_interface_network_tx_dropped`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Die Anzahl der Pakete, die übertragen werden sollten, aber von einer Netzwerkschnittstelle auf dem Knoten verworfen wurden.

`node_diskio_io_service_bytes_total`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar. Sie ist unter Windows nicht verfügbar.
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Die Gesamtzahl der Byte, die durch alle I/O-Vorgänge auf dem Knoten übertragen wurden.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`node_disk io_io_ser viced_total`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar. Sie ist unter Windows nicht verfügbar.
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Die Gesamtzahl der I/O-Vorgänge auf dem Knoten.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_cpu_reserved_capacity`	`PodName`, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
		
		`ClusterName` , `Namespace` , `Service`
			Die CPU-Kapazität, die pro Pod in einem Cluster reserviert ist.
			
			Formel: `pod_cpu_request / node_cpu_limit`
			
			Note
			`pod_cpu_request` wird nicht direkt als Metrik gemeldet, sondern ist ein Feld in Leistungsprotokoll-Ereignissen. Weitere Informationen finden Sie unter Relevante Felder in Performance-Protokollereignissen für Amazon

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			EKS und Kubernetes.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_cpu_utilization`	`PodName`, `Namespace`, `ClusterName`
	
	`Namespace`, `ClusterName`
	
	Service, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	Der Prozentsatz der CPU-Einheiten, die von Pods verwendet werden.
			
			Formel: `pod_cpu_usage_total / node_cpu_limit`
			
			Note
			`pod_cpu_usage_total` wird nicht direkt als Metrik gemeldet, sondern ist ein Feld in Leistungsprotokoll-Ereignissen. Weitere Informationen finden Sie unter Relevante Felder in Performance-Protok

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			ollereignissen für Amazon EKS und Kubernetes.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_cpu_utilization_over_pod_limit`	`PodName`, `Namespace`, `ClusterName`
	
	`Namespace`, `ClusterName`
	
	Service, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	Der Prozentsatz der CPU-Einheiten, die von Pods im Verhältnis zum Pod-Limit verwendet werden.
			
			Formel: `pod_cpu_usage_total / pod_cpu_limit`
			
			Note
			`pod_cpu_usage_total` und `pod_cpu_limit` werden nicht direkt als Metriken gemeldet, sondern sind Felder in Leistungsprotokoll-Ereignissen. Weitere Informationen finden Sie unter

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			Relevante Felder in Performance-Protokollereignissen für Amazon EKS und Kubernetes.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_memory_reserved_capacity`	`PodName`, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
		
		`ClusterName` , `Namespace` , `Service`
			Der Prozentsatz des Arbeitsspeichers, der für Pods reserviert ist.
			
			Formel: `pod_memory_request / node_memory_limit`
			
			Note
			`pod_memory_request` wird nicht direkt als Metrik gemeldet, sondern ist ein Feld in Leistungsprotokoll-Ereignissen. Weitere Informationen finden Sie unter Relevante Felder in Performance-Protok

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			ollereignissen für Amazon EKS und Kubernetes.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_memory_utilization`	`PodName`, Namespace, `ClusterName`
	
	Namespace, `ClusterName`
	
	Service, Namespace, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	Der Prozentsatz des Arbeitsspeichers, der derzeit vom Pod oder Pods verwendet wird.
			
			Formel: `pod_memory_working_set / node_memory_limit`
			
			Note
			`pod_memory_working_set` wird nicht direkt als Metrik gemeldet, sondern ist ein Feld in Leistungsprotokoll-Ereignissen. Weitere Informationen finden Sie unter Relevante Felder in

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			Performance-Protokollereignissen für Amazon EKS und Kubernetes.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_memory_utilization_over_pod_limit`	`PodName`, `Namespace`, `ClusterName`
	
	`Namespace`, `ClusterName`
	
	Service, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	Der Prozentsatz des Arbeitsspeichers, der von den Pods im Verhältnis zum Pod-Limit verwendet wird. Wenn für keinen der Container im Pod ein Speicherlimit definiert ist, wird diese Metrik nicht angezeigt.
			
			Formel: `pod_memory_working_set / pod_memory_limit`
			
			Note
			`pod_memory_working_set` wird nicht direkt als Metrik gemeldet, sondern ist ein Feld in Leistungsprotokoll-Ereignis

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			sen. Weitere Informationen finden Sie unter Relevante Felder in Performance-Protokollereignissen für Amazon EKS und Kubernetes.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_network_rx_bytes`	`PodName`, `Namespace`, `ClusterName`
	
	`Namespace`, `ClusterName`
	
	Service, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	Die Anzahl der Bytes pro Sekunde, die vom Pod über das Netzwerk empfangen werden.
			
			Formel: `sum(pod_interface_network_rx_bytes)`
			
			Note
			`pod_interface_network_rx_bytes` wird nicht direkt als Metrik gemeldet, sondern ist ein Feld in Leistungsprotokoll-Ereignissen. Weitere Informationen finden Sie unter Relevante

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			Felder in Performance-Protokollereignissen für Amazon EKS und Kubernetes.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_network_tx_bytes`	`PodName`, `Namespace`, `ClusterName`
	
	`Namespace`, `ClusterName`
	
	Service, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	Die Anzahl der Bytes pro Sekunde, die vom Pod über das Netzwerk übertragen werden.
			
			Formel: `sum(pod_interface_network_tx_bytes)`
			
			Note
			`pod_interface_network_tx_bytes` wird nicht direkt als Metrik gemeldet, sondern ist ein Feld in Leistungsprotokoll-Ereignissen. Weitere Informationen finden Sie unter Relevante

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			Felder in Performance-Protokollereignissen für Amazon EKS und Kubernetes.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_cpu_request`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Die CPU-Anfragen für den Pod.
			
			Formel: `sum(container_cpu_request)`
			
			Note
			`pod_cpu_request` wird nicht direkt als Metrik gemeldet, sondern ist ein Feld in Leistungsprotokoll-Ereignissen. Weitere Informationen finden Sie unter Relevante Felder in Performance-Protokollereignissen für Amazon

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			EKS und Kubernetes.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_memory_request`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Die Speicheranfragen für den Pod.
			
			Formel: `sum(container_memory_request)`
			
			Note
			`pod_memory_request` wird nicht direkt als Metrik gemeldet, sondern ist ein Feld in Leistungsprotokoll-Ereignissen. Weitere Informationen finden Sie unter Relevante Felder in Performance-Protokollereignissen für Amazon

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			EKS und Kubernetes.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_cpu_limit`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Das für die Container im Pod definierte CPU-Limit. Wenn für keinen der Container im Pod ein CPU-Limit definiert ist, wird diese Metrik nicht angezeigt .
			
			Formel: `sum(container_cpu_limit)`
			
			Note
			`pod_cpu_limit` wird nicht direkt als Metrik gemeldet, sondern ist ein Feld in Leistungsprotokoll-Ereignissen. Weitere Informationen finden Sie unter Relevante

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			Felder in Performance-Protokollereignissen für Amazon EKS und Kubernetes.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_memory_limit`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Das für die Container im Pod definierte Speicherlimit. Wenn für keinen der Container im Pod ein Speicherlimit definiert ist, wird diese Metrik nicht angezeigt.
			
			Formel: `sum(container_memory_limit)`
			
			Note
			`pod_cpu_limit` wird nicht direkt als Metrik gemeldet, sondern ist ein Feld in Leistungsprotokoll-Ereignissen. Weitere Informationen finden Sie unter Relevante

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			Felder in Performance-Protokollereignissen für Amazon EKS und Kubernetes.

`pod_status_failed`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Zeigt an, dass alle Container im Pod beendet wurden und mindestens ein Container mit einem Status ungleich Null beendet wurde oder vom System beendet wurde.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_status_ready`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Zeigt an, dass alle Container im Pod bereit sind, da sie den Zustand `Container Ready` erreicht haben.

`pod_status_running`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Zeigt an, dass alle Container im Pod laufen.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_status_scheduled`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Zeigt an, dass der Pod für einen Knoten geplant wurde.

`pod_status_unknown`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Zeigt an, dass der Status des Pods nicht abgerufen werden kann.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_status_pending`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Zeigt an, dass der Pod vom Cluster akzeptiert wurde, aber einer oder mehrere Container noch nicht bereit sind.

`pod_status_succeeded`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Zeigt an, dass alle Container im Pod erfolgreich beendet wurden und nicht neu gestartet werden.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_number_of_containers`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Meldet die Anzahl der Container, die in der Pod-Spezifikation definiert sind.

`pod_number_of_running_containers`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Meldet die Anzahl der Container im Pod, die sich derzeit im Status `Running` befinden.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_container_status_terminated`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Meldet die Anzahl der Container im Pod, die sich im Status `Terminated` befinden.

`pod_container_status_running`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Meldet die Anzahl der Container im Pod, die sich im Status `Running` befinden.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_container_status_waiting`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Meldet die Anzahl der Container im Pod, die sich im Status `Waiting` befinden.

`pod_interface_network_rx_dropped`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Die Anzahl der Pakete, die von einer Netzwerkschnittstelle für den Pod empfangen und anschließend verworfen wurden.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`pod_interface_network_tx_dropped`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace`, `ClusterName`
		
		`Namespace`, `ClusterName`, `Service`
		
		`ClusterName`, `Namespace`, `PodName`, `FullPodName`
			Die Anzahl der Pakete, die übertragen werden sollten, aber für den Pod verworfen wurden.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`container_cpu_utilization`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			Der Prozentsatz der CPU-Einheiten, die vom Container verwendet werden.
			
			Formel: `container_cpu_usage_total / node_cpu_limit`
			
			Note
			`container_cpu_utilization` wird nicht direkt als Metrik gemeldet, sondern ist ein Feld in Leistungsprotokoll-Ereignissen. Weitere Informationen finden Sie unter Relevante Felder in

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			Performance-Protokollereignissen für Amazon EKS und Kubernetes.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`container_cpu_utilization_over_container_limit`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			Der Prozentsatz der CPU-Einheiten, die vom Container im Verhältnis zum Container-Limit verwendet werden. Wenn für den Container kein Speicherlimit definiert ist, wird diese Metrik nicht angezeigt.
			
			Formel: `container_cpu_usage_total / container_cpu_limit`
			
			Note
			`container_cpu_utilization_over_container_limit` wird nicht direkt als Metrik gemeldet, sondern ist

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			ein Feld in Leistungsprotokoll-Ereignissen. Weitere Informationen finden Sie unter Relevante Felder in Performance-Protokollereignissen für Amazon EKS und Kubernetes.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`container_memory_utilization`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			Prozentsatz der Speichereinheiten, die vom Container verwendet werden.
			
			Formel: `container_memory_working_set / node_memory_limit`
			
			Note
			`container_memory_utilization` wird nicht direkt als Metrik gemeldet, sondern ist ein Feld in Leistungsprotokoll-Ereignissen. Weitere Informationen finden Sie unter Relevante

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			Felder in Performance-Protokollereignissen für Amazon EKS und Kubernetes.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`container_memory_utilization_over_container_limit`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			Prozentsatz der vom Container benutzten Speichereinheiten im Verhältnis zum Container-Limit. Wenn für den Container kein Speicherlimit definiert ist, wird diese Metrik nicht angezeigt.
			
			Formel: `container_memory_working_set / container_memory_limit`
			
			Note
			`container_memory_utilization_over_container_limit` wird nicht direkt als Metrik gemeldet, sondern ist ein Feld in Leistungs

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
			protokoll -Ereignis sen. Weitere Informati onen finden Sie unter Relevante Felder in Performan ce-Protok ollereignissen für Amazon EKS und Kubernetes.

`container_memory_failures_total`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar. Sie ist unter Windows nicht verfügbar.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			Die Anzahl der Fehler bei der Speicherz uweisung, die beim Container aufgetreten sind.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung
`pod_number_of_container_restarts`	PodName, `Namespace` , `ClusterName`		Die Gesamtanzahl der Container-Neustarts in einem Pod.
`service_number_of_running_pods`	Service `Namespace` , `ClusterName` `ClusterName`		Die Anzahl der Pods, von denen der Service oder die Services im Cluster ausgeführt werden.
`replicas_desired` Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar		`ClusterName` PodName, `Namespace` , `ClusterName`	Die Anzahl der Pods, die für einen Workload gewünscht werden, wie in der Workload-Spezifikation definiert.
`replicas_ready` Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar		`ClusterName` PodName, `Namespace` , `ClusterName`	Die Anzahl der Pods für einen Workload, die den Status Bereit erreicht haben.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`status_replicas_available`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
			Die Anzahl der verfügbaren Pods für einen Workload. Ein Pod ist verfügbar, wenn er für die in der Workload-Spezifikation definierten `minReadySeconds` bereit ist.

`status_replicas_unavailable`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
			Die Anzahl der Pods für einen Workload, die nicht verfügbar sind. Ein Pod ist verfügbar, wenn er für die in der Workload-Spezifikation definierten `minReadySeconds` bereit ist. Pods sind nicht verfügbar, wenn sie dieses Kriterium nicht erfüllen.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`apiserver_storage_objects`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName`, `resource`
			Die Anzahl der zum Zeitpunkt der letzten Prüfung in etcd gespeicherten Objekte.

`apiserver_request_total`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName`, `code`, `verb`
			Die Gesamtzahl der API-Anfragen an den Kubernetes-API-Server.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`apiserver_request_duration_seconds`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName , verb`
			Reaktionslatenz für API-Anfragen an den Kubernetes-API-Server.

`apiserver_admission_controller_admission_duration_seconds`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName , operation`
			Latenz des Admission Controllers in Sekunden. Ein Admission Controller ist Code, der Anfragen an den Kubernetes-API-Server abfängt.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`rest_client_request_duration_seconds`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName`, `operation`
			Reaktionslatenz bei Clients, die den Kubernetes-API-Server aufrufen. Diese Metrik ist experimentell und kann sich in zukünftigen Versionen von Kubernetes ändern.

`rest_client_requests_total`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName`, `code`, `method`
			Die Gesamtzahl der API-Anfragen, die von Clients an den Kubernetes-API-Server gestellt wurden. Diese Metrik ist experimentell und kann sich in zukünftigen Versionen von Kubernetes ändern.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`etcd_request_duration_seconds`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName , operation`
			Antwortlatenz bei API-Aufrufen an Etcd. Diese Metrik ist experimentell und kann sich in zukünftigen Versionen von Kubernetes ändern.

`apiserver_storage_size_bytes`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName , endpoint`
			Größe der physisch zugewiesenen Speicherdatenbankdatei in Byte. Diese Metrik ist experimentell und kann sich in zukünftigen Versionen von Kubernetes ändern.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`apiserver_longrunning_requests`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName`, `resource`
			Die Anzahl der aktiven Anfragen mit langer Laufzeit an den Kubernetes-API-Server.

`apiserver_current_inflight_requests`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName`, `request_kind`
			Die Anzahl der Anfragen, die vom Kubernetes-API-Server verarbeitet werden.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`apiserver_admission_webhook_admission_duration_seconds`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName`, `name`
			Webhook-Latenz bei der Zulassung in Sekunden. Zulassungs-Webhooks sind HTTP-Callbacks, die Zulassungsanfragen empfangen und etwas damit anfangen.

`apiserver_admission_step_admission_duration_seconds`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName`, `operation`
			Latenz der Teilschritte bei der Zulassung in Sekunden.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`apiserver_requested_deprecated_apis`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName`, `group`
			Anzahl der Anfragen an veraltete APIs auf dem Kubernetes-API-Server.

`apiserver_request_total_5XX`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName`, `code`, `verb`
			Anzahl der Anfragen an den Kubernetes-API-Server, auf die mit einem 5XX-HTTP-Antwortcode geantwortet wurde.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`apiserver_storage_list_duration_seconds`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName , resource`
			Reaktionslatenz beim Auflisten von Objekten aus Etcd. Diese Metrik ist experimentell und kann sich in zukünftigen Versionen von Kubernetes ändern.

`apiserver_current_inqueue_requests`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName , request_kind`
			Die Anzahl der Anfragen in der Warteschlange, die vom Kubernetes-API-Server in die Warteschlange gestellt wurden. Diese Metrik ist experimentell und kann sich in zukünftigen Versionen von Kubernetes ändern.

Metrikname	Dimensionen mit jeder Version von Container Insights	Zusätzliche Dimensionen mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	Beschreibung

`apiserver_flowcontrol_rejected_requests_total`
 	
Diese Metrik ist nur mit Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS verfügbar
		`ClusterName`
		
		`ClusterName , reason`
			Anzahl der Anfragen, die vom API-Subsystem Priority and Fairness abgelehnt wurden. Diese Metrik ist experimentell und kann sich in zukünftigen Versionen von Kubernetes ändern.

NVIDIA GPU-Metriken

Ab `1.300034.0` der Version des CloudWatch Agenten erfasst Container Insights mit verbesserter Observability für Amazon EKS standardmäßig NVIDIA-GPU-Metriken von EKS-Workloads. Der CloudWatch Agent muss mit der Add-On-Version `v1.3.0-eksbuild.1` von CloudWatch Observability EKS oder höher installiert werden. Weitere Informationen finden Sie unter Installieren Sie den CloudWatch Agenten mithilfe des Amazon CloudWatch Observability EKS-Add-ons. Diese gesammelten NVIDIA-GPU-Metriken sind in der Tabelle in diesem Abschnitt aufgeführt.

Damit Container Insights NVIDIA-GPU-Metriken erfassen kann, müssen Sie die folgenden Voraussetzungen erfüllen:

- Sie müssen Container Insights mit erweiterter Observability für Amazon EKS mit der Amazon CloudWatch Observability EKS-Add-On-Version `v1.3.0-eksbuild.1` oder höher verwenden.
- Das NVIDIA-Geräte-Plugin für Kubernetes muss im Cluster installiert sein.

- Das NVIDIA-Container-Toolkit muss auf den Knoten des Clusters installiert sein. Beispielsweise werden die für Amazon EKS optimierten beschleunigten AMIs mit den erforderlichen Komponenten erstellt.

Sie können die Erfassung von NVIDIA-GPU-Metriken deaktivieren, indem Sie die `accelerated_compute_metrics` Option in der CloudWatch Start-Agent-Konfigurationsdatei auf einstellen`false`. Weitere Informationen und ein Beispiel für eine Opt-Out-Konfiguration finden Sie unter(Optional) Zusätzliche Konfiguration.

Metrikname	Dimensionen	Beschreibung

`container_gpu_memory_total`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `GpuDevice`
		Die Gesamtgröße des Frame-Buffers in Byte auf den GPU (s), die dem Container zugewiesen sind.

`container_gpu_memory_used`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `GpuDevice`
		Die Byte des Frame-Buffers, die auf den GPU (s) verwendet werden, die dem Container zugewiesen sind.

Metrikname	Dimensionen	Beschreibung
container_gpu_memory_utilization	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	Der Prozentsatz des verwendeten Frame-Buffers von den GPU (s), die dem Container zugewiesen sind.
container_gpu_power_draw	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	Der Stromverbrauch der GPU (s), die dem Container zugewiesen sind, in Watt.

Metrikname	Dimensionen	Beschreibung
container_gpu_temperature	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	Die Temperatur der GPU (s), die dem Container zugewiesen sind, in Grad Celsius.
container_gpu_utilization	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	Die prozentuale Auslastung der GPU (s), die dem Container zugewiesen sind.
node_gpu_memory_total	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	Die Gesamtgröße des Frame-Buffers in Byte auf den GPU (s), die dem Knoten zugewiesen sind.

Metrikname	Dimensionen	Beschreibung
node_gpu_memory_used	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	Die Byte des Frame-Buffers, der auf den GPU (s) verwendet wird, die dem Knoten zugewiesen sind.
node_gpu_memory_utilization	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	Der Prozentsatz des Frame-Buffers, der auf den GPU (s) verwendet wird, die dem Knoten zugewiesen sind.
node_gpu_power_draw	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	Der Stromverbrauch der GPU (s), die dem Knoten zugewiesen sind, in Watt.
node_gpu_temperature	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	Die Temperatur der dem Knoten zugewiesenen GPU (s) in Grad Celsius.

Metrikname	Dimensionen	Beschreibung
node_gpu_utilization	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	Die prozentuale Auslastung der GPU (s), die dem Knoten zugewiesen sind.
pod_gpu_memory_total	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	Die Gesamtgröße des Frame-Buffers in Byte auf den GPU (s), die dem Pod zugewiesen sind.

Metrikname	Dimensionen	Beschreibung
pod_gpu_memory_used	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	Die Byte des Frame-Buffers, der auf den GPU (s) verwendet wird, die dem Pod zugewiesen sind.
pod_gpu_memory_utilization	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	Der Prozentsatz des verwendeten Frame-Buffers der GPU (s), die dem Pod zugewiesen sind.

Metrikname	Dimensionen	Beschreibung
pod_gpu_power_draw	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	Der Stromverbrauch der GPU (s), die dem Pod zugewiesen sind, in Watt.
pod_gpu_temperature	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	Die Temperatur der GPU (s), die dem Pod zugewiesen sind, in Grad Celsius.

Metrikname	Dimensionen	Beschreibung
pod_gpu_utilization	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	Die prozentuale Auslastung der GPU (s), die dem Pod zugewiesen sind.

AWS Neuronenmetriken für AWS Trainium und Inferentia AWS

Ab `1.300036.0` der Version des CloudWatch Agenten erfasst Container Insights mit verbesserter Observability für Amazon EKS standardmäßig beschleunigte Rechenmetriken von AWS Trainium- und AWS Inferentia-Beschleunigern. Der CloudWatch Agent muss mit der Add-On-Version von CloudWatch Observability EKS oder höher installiert werden. `v1.5.0-eksbuild.1` Weitere Informationen zum Add-on finden Sie unterInstallieren Sie den CloudWatch Agenten mithilfe des Amazon CloudWatch Observability EKS-Add-ons. Weitere Informationen zu AWS Trainium finden Sie AWS unter Trainium. Weitere Informationen zu Inferentia finden Sie unter AWS Inferentia.AWS

Damit Container Insights AWS Neuron-Metriken sammeln kann, müssen Sie die folgenden Voraussetzungen erfüllen:

- Sie müssen Container Insights mit erweiterter Observability für Amazon EKS mit der Amazon CloudWatch Observability EKS-Add-On-Version `v1.5.0-eksbuild.1` oder höher verwenden.
- Der Neuron-Treiber muss auf den Knoten des Clusters installiert sein.
- Das Neuron-Geräte-Plugin muss auf dem Cluster installiert sein. Beispielsweise werden die für Amazon EKS optimierten beschleunigten AMIs mit den erforderlichen Komponenten erstellt.

Die gesammelten Metriken sind in der Tabelle in diesem Abschnitt aufgeführt. Die Metriken werden für AWS Trainium, AWS Inferentia und Inferentia2 gesammelt. AWS

Der CloudWatch Agent sammelt diese Metriken vom Neuron-Monitor und führt die erforderliche Korrelation der Kubernetes-Ressourcen durch, um Metriken auf Pod- und Container-Ebene bereitzustellen

Metrikname	Dimensionen	Beschreibung
container_neuroncore_utilization	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	NeuronCore Nutzung der dem Container NeuronCore zugewiesenen Daten während des erfassten Zeitraums. Einheit: Prozent
container_neuroncore_memory_usage_constants	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	Die Menge an Gerätespeicher, die während des Trainings von dem NeuronCore , der dem Container zugewiesen wurde, für Konstanten verwendet wird (oder für Gewichte während der Inferenz). Einheit: Byte
container_neuroncore_memory_usage_model_code	ClusterName ClusterName , Namespace , PodName, ContainerName	Die Größe des Gerätespeichers, der von dem, der dem Container zugewiesen ist NeuronCore , für den ausführbaren Code der Modelle verwendet wird.

Metrikname	Dimensionen	Beschreibung

	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `NeuronDevice` , `NeuronCore`
		Einheit: Byte

`container_neuroncore_memory_usage_model_shared_scratchpad`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `NeuronDevice` , `NeuronCore`
		Die Größe des Gerätespeichers, der für das Scratchpad verwendet wird, das von den Modellen gemeinsam genutzt wird NeuronCore , das dem Container zugewiesen ist. Dieser Speicherbereich ist für die Modelle reserviert.
		
		Einheit: Byte

`container_neuroncore_memory_usage_runtime_memory`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `NeuronDevice` , `NeuronCore`
		Die Menge an Gerätespeicher, die für die Neuron-Laufzeit von den dem Container NeuronCore zugewiesenen Geräten verwendet wird.
		
		Einheit: Byte

Metrikname	Dimensionen	Beschreibung
container_neuroncore_memory_usage_tensors	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	Die Größe des Gerätespeichers, der von dem dem Container NeuronCore zugewiesenen Gerät für Tensoren verwendet wird. Einheit: Byte
container_neuroncore_memory_usage_total	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	Die Gesamtmenge an Speicher, die von dem dem Container NeuronCore zugewiesenen Speicherplatz verwendet wird. Einheit: Byte

Metrikname	Dimensionen	Beschreibung
container_neurondevice_hw_ecc_events_total	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice	Die Anzahl der korrigierten und unkorrigierten ECC-Ereignisse für das chipinterne SRAM und den Gerätespeicher des Neuron-Geräts auf dem Knoten. Einheit: Anzahl
pod_neuroncore_utilization	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	Die NeuronCore Nutzung der dem Pod zugewiesenen Daten während des NeuronCore erfassten Zeitraums . Einheit: Prozent

Metrikname	Dimensionen	Beschreibung
pod_neuroncore_memory_usage_constants	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	Die Größe des Gerätespeichers, der während des Trainings für Konstanten verwendet wird NeuronCore , die dem Pod zugewiesen wurden (oder für Gewichte während der Inferenz). Einheit: Byte
pod_neuroncore_memory_usage_model_code	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	Die Größe des Gerätespeichers, der von dem, der dem Pod zugewiesen wurde NeuronCore , für den ausführbaren Code der Modelle verwendet wird. Einheit: Byte

Metrikname	Dimensionen	Beschreibung
pod_neuroncore_memory_usage_model_shared_scratchpad	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	Die Größe des Gerätespeichers, der für das Scratchpad verwendet wird, das von den Modellen gemeinsam genutzt wird NeuronCore , das dem Pod zugewiesen ist. Dieser Speicherbereich ist für die Modelle reserviert. Einheit: Byte
pod_neuroncore_memory_usage_runtime_memory	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	Die Menge an Gerätespeicher, die für die Neuron-Laufzeit von dem dem Pod NeuronCore zugewiesenen Gerät verwendet wird. Einheit: Byte

Metrikname	Dimensionen	Beschreibung
pod_neuroncore_memory_usage_tensors	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	Die Größe des Gerätespeichers, der von dem dem Pod NeuronCore zugewiesenen Gerät für Tensoren verwendet wird. Einheit: Byte
pod_neuroncore_memory_usage_total	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	Die Gesamtmenge des Speichers , der von dem dem Pod NeuronCore zugewiesenen Speicherplatz verwendet wird. Einheit: Byte

Metrikname	Dimensionen	Beschreibung
pod_neurondevice_hw_ecc_events_total	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice	Die Anzahl der korrigierten und unkorrigierten ECC-Ereignisse für das chipinterne SRAM und den Gerätespeicher des Neuron-Geräts, das einem Pod zugewiesen ist. Einheit: Byte
node_neuroncore_utilization	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	Die NeuronCore Auslastung des dem Knoten zugewiesenen Zeitraums während des NeuronCore erfassten Zeitraums. Einheit: Prozent
node_neuroncore_memory_usage_constants	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	Die Menge des Gerätespeichers, der während des Trainings für Konstanten verwendet wird NeuronCore , die dem Knoten zugewiesen wurden (oder Gewichte während der Inferenz). Einheit: Byte

Metrikname	Dimensionen	Beschreibung
node_neuroncore_memory_usage_model_code	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	Die Größe des Gerätespeichers, der von dem, der dem Knoten zugewiesen ist NeuronCore , für den ausführbaren Code der Modelle verwendet wird. Einheit: Byte
node_neuroncore_memory_usage_model_shared_scratchpad	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	Die Größe des Gerätespeichers, der für das Scratchpad verwendet wird, das von den Modellen gemeinsam genutzt wird NeuronCore , das dem Knoten zugewiesen ist. Dies ist ein Speicherbereich, der für die Modelle reserviert ist. Einheit: Byte
node_neuroncore_memory_usage_runtime_memory	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	Die Menge an Gerätespeicher, die für die Neuron-Laufzeit von dem verwendet wird NeuronCore , der dem Knoten zugewiesen ist. Einheit: Byte
node_neuroncore_memory_usage_tensors	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	Die Menge des Gerätespeichers, der von dem für Tensoren verwendet wird NeuronCore , der dem Knoten zugewiesen ist. Einheit: Byte

Metrikname	Dimensionen	Beschreibung
node_neuroncore_memory_usage_total	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	Die Gesamtmenge des Speichers , der von dem verwendet wird NeuronCore , der dem Knoten zugewiesen ist. Einheit: Byte
node_neuron_execution_errors_total	ClusterName ClusterName , InstanceId , NodeName	Die Gesamtzahl der Ausführungsfehler auf dem Knoten. Dies wird vom CloudWatch Agenten berechnet, indem er die Fehler der folgenden Typen aggregiert:generic,numerical ,transient , modelruntime, und hardware Einheit: Anzahl
node_neurondevice_runtime_memory_used_bytes	ClusterName ClusterName , InstanceId , NodeName	Die gesamte Speichernutzung des Neuron-Geräts auf dem Knoten in Byte. Einheit: Byte
node_neuron_execution_latency	ClusterName ClusterName , InstanceId , NodeName	In Sekunden, die Latenz für eine Ausführung auf dem Knoten, gemessen anhand der Neuron-Laufzeit. Einheit: Sekunden

Metrikname	Dimensionen	Beschreibung

`node_neuron device_hw_ecc_events_total`	`ClusterName`
	
	`ClusterName` , `InstanceId` , `NodeName`
	
	`ClusterName` , `InstanceId` , `NodeName`, `NeuronDevice`
		Die Anzahl der korrigierten und unkorrigierten ECC-Ereignisse für das chipinterne SRAM und den Gerätespeicher des Neuron-Geräts auf dem Knoten.
		
		Einheit: Anzahl

AWS Metriken für Elastic Fabric Adapter (EFA)

Ab `1.300037.0` der Version des CloudWatch Agenten sammelt Container Insights mit verbesserter Observability für Amazon EKS AWS Elastic Fabric Adapter (EFA) -Metriken aus Amazon EKS-Clustern auf Linux-Instances. Der CloudWatch Agent muss mit der Add-On-Version `v1.5.2-eksbuild.1` von CloudWatch Observability EKS oder höher installiert werden. Weitere Informationen zum Add-on finden Sie unterInstallieren Sie den CloudWatch Agenten mithilfe des Amazon CloudWatch Observability EKS-Add-ons. Weitere Informationen zum AWS Elastic Fabric Adapter finden Sie unter Elastic Fabric Adapter.

Damit Container Insights Metriken für AWS Elastic Fabric-Adapter sammeln kann, müssen Sie die folgenden Voraussetzungen erfüllen:

- Sie müssen Container Insights mit erweiterter Observability für Amazon EKS mit der Amazon CloudWatch Observability EKS-Add-On-Version `v1.5.2-eksbuild.1` oder höher verwenden.
- Das EFA-Geräte-Plugin muss auf dem Cluster installiert sein. Weitere Informationen finden Sie unter aws-efa-ks-device-plugin8. GitHub

Die gesammelten Metriken sind in der folgenden Tabelle aufgeführt.

Metrikname	Dimensionen	Beschreibung

`container_efa_rx_bytes`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
		Die Anzahl der Byte pro Sekunde, die von den dem Container zugewiesenen EFA-Geräten empfangen wurden.

Metrikname	Dimensionen	Beschreibung

	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `EfaDevice`
		Einheit: Byte/Sekunde

`container_efa_tx_bytes`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `EfaDevice`
		Die Anzahl der Byte pro Sekunde, die von den dem Container zugewiesenen EFA-Geräten übertragen werden.
		
		Einheit: Byte/Sekunde

`container_efa_rx_dropped`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `EfaDevice`
		Die Anzahl der Pakete, die von den dem Container zugewiesenen EFA-Geräten empfangen und dann verworfen wurden.
		
		Einheit: Zählung/Sekunde

Metrikname	Dimensionen	Beschreibung
container_efa_rdma_read_bytes	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , EfaDevice	Die Anzahl der Byte pro Sekunde, die mithilfe von Lesevorgängen mit direktem Fernzugriff auf den Speicher von den dem Container zugewiesenen EFA-Geräten empfangen wurden. Einheit: Byte/Sekunde
container_efa_rdma_write_bytes	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , EfaDevice	Die Anzahl der Byte pro Sekunde, die mithilfe von Lesevorgängen mit direktem Fernzugriff auf den Speicher durch die dem Container zugewiesenen EFA-Geräte übertragen wurden. Einheit: Byte/Sekunde

Metrikname	Dimensionen	Beschreibung
container_efa_rdma_write_recv_bytes	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , EfaDevice	Die Anzahl der Byte pro Sekunde, die bei Schreibvorgängen mit direktem Fernzugriff auf den Speicher von den dem Container zugewiesenen EFA-Geräten empfangen wurden. Einheit: Byte/Sekunde
pod_efa_rx_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	Die Anzahl der Byte pro Sekunde, die von den dem Pod zugewiesenen EFA-Geräten empfangen wurden. Einheit: Byte/Sekunde

Metrikname	Dimensionen	Beschreibung
pod_efa_tx_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	Die Anzahl der Byte pro Sekunde, die von den dem Pod zugewiesenen EFA-Geräten übertragen werden. Einheit: Byte/Sekunde
pod_efa_rx_dropped	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	Die Anzahl der Pakete, die von den dem Pod zugewiesenen EFA-Gerät en empfangen und dann verworfen wurden. Einheit: Zählung/Sekunde

Metrikname	Dimensionen	Beschreibung
pod_efa_rdma_read_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	Die Anzahl der Byte pro Sekunde, die mithilfe von Lesevorgängen mit direktem Fernzugriff auf den Speicher von den dem Pod zugewiesenen EFA-Geräten empfangen wurden. Einheit: Byte/Sekunde
pod_efa_rdma_write_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	Die Anzahl der Byte pro Sekunde, die mithilfe von Lesevorgängen mit direktem Fernzugriff auf den Speicher durch die dem Pod zugewiesenen EFA-Geräte übertragen wurden. Einheit: Byte/Sekunde

Metrikname	Dimensionen	Beschreibung
pod_efa_rdma_write_recv_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	Die Anzahl der Byte pro Sekunde, die bei Schreibvorgängen mit direktem Fernzugriff auf den Speicher von den dem Pod zugewiesenen EFA-Gerät en empfangen wurden. Einheit: Byte/Sekunde
node_efa_rx_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	Die Anzahl der Byte pro Sekunde, die von den dem Knoten zugewiesenen EFA-Geräten empfangen wurden. Einheit: Byte/Sekunde
node_efa_tx_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	Die Anzahl der Byte pro Sekunde, die von den dem Knoten zugewiesenen EFA-Geräten übertragen werden. Einheit: Byte/Sekunde

Metrikname	Dimensionen	Beschreibung
node_efa_rx_dropped	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	Die Anzahl der Pakete, die von den dem Knoten zugewiesenen EFA-Geräten empfangen und dann verworfen wurden. Einheit: Zählung/Sekunde
node_efa_rdma_read_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	Die Anzahl der Byte pro Sekunde, die mithilfe von Lesevorgängen mit direktem Fernzugriff auf den Speicher von den dem Knoten zugewiesenen EFA-Geräten empfangen wurden. Einheit: Byte/Sekunde
pod_efa_rdma_write_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	Die Anzahl der Byte pro Sekunde, die mithilfe von Lesevorgängen mit direktem Fernzugriff auf den Speicher durch die dem Pod zugewiesenen EFA-Geräte übertragen wurden. Einheit: Byte/Sekunde
node_efa_rdma_write_recv_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	Die Anzahl der Byte pro Sekunde, die bei Schreibvorgängen mit direktem Fernzugriff auf den Speicher von den dem Knoten zugewiesenen EFA-Geräten empfangen wurden. Einheit: Byte/Sekunde

Referenz zu Container-Insights-Leistungsprotokollen

Dieser Abschnitt enthält Referenzinformationen darüber, wie Container Insights Peformance-Protokollereignisse verwendet, um Metriken zu erfassen. Wenn Sie Container Insights einsetzen, erstellt es automatisch eine Protokollgruppe für die Leistungsprotokoll-Ereignisse. Sie müssen diese Protokollgruppe nicht selbst erstellen.

Themen

Performance-Protokollereignisse von Container Insights für Amazon ECS

Im Folgenden finden Sie Beispiele für die Performance-Protokollereignisse, die Container Insights von Amazon ECS sammelt.

Diese Protokolle befinden sich in der Protokollgruppe CloudWatch Logs mit dem Namen`/aws/ecs/containerinsights/`*`CLUSTER_NAME`*`/performance`. Innerhalb dieser Protokollgruppe hat jede Container-Instance einen Protokollstream mit dem Namen `AgentTelemetry-`*`CONTAINER_INSTANCE_ID`*.

Sie können diese Protokolle mithilfe von Abfragen abfragen, z. B. `{ $.Type = "Container" }`, um alle Container-Protokollereignisse anzuzeigen.

Typ: Container

```
{
 "Version":"0",
 "Type":"Container",
 "ContainerName":"sleep",
 "TaskId":"7ac4dfba69214411b4783a3b8189c9ba",
 "TaskDefinitionFamily":"sleep360",
 "TaskDefinitionRevision":"1",
 "ContainerInstanceId":"0d7650e6dec34c1a9200f72098071e8f",
 "EC2InstanceId":"i-0c470579dbcdbd2f3",
 "ClusterName":"MyCluster",
 "Image":"busybox",
 "ContainerKnownStatus":"RUNNING",
```

```
 "Timestamp":1623963900000,
 "CpuUtilized":0.0,
 "CpuReserved":10.0,
 "MemoryUtilized":0,
 "MemoryReserved":10,
 "StorageReadBytes":0,
 "StorageWriteBytes":0,
 "NetworkRxBytes":0,
 "NetworkRxDropped":0,
 "NetworkRxErrors":0,
 "NetworkRxPackets":14,
 "NetworkTxBytes":0,
 "NetworkTxDropped":0,
 "NetworkTxErrors":0,
 "NetworkTxPackets":0
}
```

Typ: Aufgabe

```
{
    "Version": "0",
    "Type": "Task",
    "TaskId": "7ac4dfba69214411b4783a3b8189c9ba",
    "TaskDefinitionFamily": "sleep360",
    "TaskDefinitionRevision": "1",
    "ContainerInstanceId": "0d7650e6dec34c1a9200f72098071e8f",
    "EC2InstanceId": "i-0c470579dbcdbd2f3",
    "ClusterName": "MyCluster",
    "AccountID": "637146863587",
    "Region": "us-west-2",
    "AvailabilityZone": "us-west-2b",
    "KnownStatus": "RUNNING",
    "LaunchType": "EC2",
    "PullStartedAt": 1623963608201,
    "PullStoppedAt": 1623963610065,
    "CreatedAt": 1623963607094,
    "StartedAt": 1623963610382,
    "Timestamp": 1623963900000,
    "CpuUtilized": 0.0,
    "CpuReserved": 10.0,
    "MemoryUtilized": 0,
    "MemoryReserved": 10,
    "StorageReadBytes": 0,
```

```
    "StorageWriteBytes": 0,
    "NetworkRxBytes": 0,
    "NetworkRxDropped": 0,
    "NetworkRxErrors": 0,
    "NetworkRxPackets": 14,
    "NetworkTxBytes": 0,
    "NetworkTxDropped": 0,
    "NetworkTxErrors": 0,
    "NetworkTxPackets": 0,
    "EBSFilesystemUtilized": 10,
    "EBSFilesystemSize": 20,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "CpuUtilized",
                    "Unit": "None"
                },
                {
                    "Name": "CpuReserved",
                    "Unit": "None"
                },
                {
                    "Name": "MemoryUtilized",
                    "Unit": "Megabytes"
                },
                {
                    "Name": "MemoryReserved",
                    "Unit": "Megabytes"
                },
                {
                    "Name": "StorageReadBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "StorageWriteBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "NetworkRxBytes",
                    "Unit": "Bytes/Second"
                },
                {
```

```
                    "Name": "NetworkTxBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "EBSFilesystemSize",
                    "Unit": "Gigabytes"
                },
                {
                    "Name": "EBSFilesystemUtilzed",
                    "Unit": "Gigabytes"
                }
            ],
            "Dimensions": [
                ["ClusterName"],
                [
                    "ClusterName",
                    "TaskDefinitionFamily"
                ]
            ]
        }
    ]
}
```

Typ: Service

```
{
    "Version": "0",
    "Type": "Service",
    "ServiceName": "myCIService",
    "ClusterName": "myCICluster",
    "Timestamp": 1561586460000,
    "DesiredTaskCount": 2,
    "RunningTaskCount": 2,
    "PendingTaskCount": 0,
    "DeploymentCount": 1,
    "TaskSetCount": 0,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "DesiredTaskCount",
                    "Unit": "Count"
```

```
                },
                {
                    "Name": "RunningTaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "PendingTaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "DeploymentCount",
                    "Unit": "Count"
                },
                {
                    "Name": "TaskSetCount",
                    "Unit": "Count"
                }
            ],
            "Dimensions": [
                [
                    "ServiceName",
                    "ClusterName"
                ]
            ]
        }
    ]
}
```

Typ: Volumen

```
{
    "Version": "0",
    "Type": "Volume",
    "TaskDefinitionFamily": "myCITaskDef",
    "TaskId": "7ac4dfba69214411b4783a3b8189c9ba",
    "ClusterName": "myCICluster",
    "ServiceName": "myCIService",
    "VolumeId": "vol-1233436545ff708cb",
    "InstanceId": "i-0c470579dbcdbd2f3",
    "LaunchType": "EC2",
    "VolumeName": "MyVolumeName",
    "EBSFilesystemUtilized": 10,
    "EBSFilesystemSize": 20,
```

```
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "EBSFilesystemSize",
                    "Unit": "Gigabytes"
                },
                {
                    "Name": "EBSFilesystemUtilzed",
                    "Unit": "Gigabytes"
                }
            ],
            "Dimensions": [
                ["ClusterName"],
                [
                    "VolumeName",
                    "TaskDefinitionFamily",
                    "ClusterName"
                ],
                [
                    "ServiceName",
                    "ClusterName"
                ]
            ]
        }
    ]
}
```

Typ: Cluster

```
{
    "Version": "0",
    "Type": "Cluster",
    "ClusterName": "myCICluster",
    "Timestamp": 1561587300000,
    "TaskCount": 5,
    "ContainerInstanceCount": 5,
    "ServiceCount": 2,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
```

```
                {
                    "Name": "TaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "ContainerInstanceCount",
                    "Unit": "Count"
                },
                {
                    "Name": "ServiceCount",
                    "Unit": "Count"
                }
            ],
            "Dimensions": [
                [
                    "ClusterName"
                ]
            ]
        }
    ]
}
```

Container Insights-Performance-Protokollereignisse für Amazon EKS und Kubernetes

Im Folgenden finden Sie Beispiele für die Performance-Protokollereignisse, die Container Insights von Amazon-EKS- und Kubernetes-Clustern sammelt.

Typ: Knoten

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "node_cpu_utilization"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_utilization"
        },
```

```
        {
          "Unit": "Bytes/Second",
          "Name": "node_network_total_bytes"
        },
        {
          "Unit": "Percent",
          "Name": "node_cpu_reserved_capacity"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_reserved_capacity"
        },
        {
          "Unit": "Count",
          "Name": "node_number_of_running_pods"
        },
        {
          "Unit": "Count",
          "Name": "node_number_of_running_containers"
        }
      ],
      "Dimensions": [
        [
          "NodeName",
          "InstanceId",
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    },
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "node_cpu_utilization"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_utilization"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "node_network_total_bytes"
        },
```

```
        {
          "Unit": "Percent",
          "Name": "node_cpu_reserved_capacity"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_reserved_capacity"
        },
        {
          "Unit": "Count",
          "Name": "node_number_of_running_pods"
        },
        {
          "Unit": "Count",
          "Name": "node_number_of_running_containers"
        },
        {
          "Name": "node_cpu_usage_total"
        },
        {
          "Name": "node_cpu_limit"
        },
        {
          "Unit": "Bytes",
          "Name": "node_memory_working_set"
        },
        {
          "Unit": "Bytes",
          "Name": "node_memory_limit"
        }
      ],
      "Dimensions": [
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
```

```
    "cadvisor",
    "/proc",
    "pod",
    "calculated"
  ],
  "Timestamp": "1567096682364",
  "Type": "Node",
  "Version": "0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_cpu_limit": 4000,
  "node_cpu_request": 1130,
  "node_cpu_reserved_capacity": 28.249999999999996,
  "node_cpu_usage_system": 33.794636630852764,
  "node_cpu_usage_total": 136.47852169244098,
  "node_cpu_usage_user": 71.67075111567326,
  "node_cpu_utilization": 3.4119630423110245,
  "node_memory_cache": 3103297536,
  "node_memory_failcnt": 0,
  "node_memory_hierarchical_pgfault": 0,
  "node_memory_hierarchical_pgmajfault": 0,
  "node_memory_limit": 16624865280,
  "node_memory_mapped_file": 406646784,
  "node_memory_max_usage": 4230746112,
  "node_memory_pgfault": 0,
  "node_memory_pgmajfault": 0,
  "node_memory_request": 1115684864,
  "node_memory_reserved_capacity": 6.7109407818311055,
  "node_memory_rss": 798146560,
  "node_memory_swap": 0,
  "node_memory_usage": 3901444096,
  "node_memory_utilization": 6.601302600149552,
  "node_memory_working_set": 1097457664,
  "node_network_rx_bytes": 35918.392817386324,
  "node_network_rx_dropped": 0,
  "node_network_rx_errors": 0,
  "node_network_rx_packets": 157.67565245448117,
  "node_network_total_bytes": 68264.20276554905,
  "node_network_tx_bytes": 32345.80994816272,
  "node_network_tx_dropped": 0,
  "node_network_tx_errors": 0,
  "node_network_tx_packets": 154.21455923431654,
  "node_number_of_running_containers": 16,
```

```
  "node_number_of_running_pods": 13
}
```

Typ: NodeFS

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "node_filesystem_utilization"
        }
      ],
      "Dimensions": [
        [
          "NodeName",
          "InstanceId",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "EBSVolumeId": "aws://us-west-2b/vol-0a53108976d4a2fda",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567097939726",
  "Type": "NodeFS",
  "Version": "0",
  "device": "/dev/nvme0n1p1",
  "fstype": "vfs",
```

```
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_filesystem_available": 17298395136,
  "node_filesystem_capacity": 21462233088,
  "node_filesystem_inodes": 10484720,
  "node_filesystem_inodes_free": 10367158,
  "node_filesystem_usage": 4163837952,
  "node_filesystem_utilization": 19.400767547940255
}
```

Typ: NodeDisk IO

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "EBSVolumeId": "aws://us-west-2b/vol-0a53108976d4a2fda",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor"
  ],
  "Timestamp": "1567096928131",
  "Type": "NodeDiskIO",
  "Version": "0",
  "device": "/dev/nvme0n1",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_diskio_io_service_bytes_async": 9750.505814277016,
  "node_diskio_io_service_bytes_read": 0,
  "node_diskio_io_service_bytes_sync": 230.6174506688036,
  "node_diskio_io_service_bytes_total": 9981.123264945818,
  "node_diskio_io_service_bytes_write": 9981.123264945818,
  "node_diskio_io_serviced_async": 1.153087253344018,
  "node_diskio_io_serviced_read": 0,
  "node_diskio_io_serviced_sync": 0.03603397666700056,
  "node_diskio_io_serviced_total": 1.1891212300110185,
  "node_diskio_io_serviced_write": 1.1891212300110185
}
```

Typ: NodeNet


```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567096928131",
  "Type": "NodeNet",
  "Version": "0",
  "interface": "eni972f6bfa9a0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_interface_network_rx_bytes": 3163.008420864309,
  "node_interface_network_rx_dropped": 0,
  "node_interface_network_rx_errors": 0,
  "node_interface_network_rx_packets": 16.575629266820258,
  "node_interface_network_total_bytes": 3518.3935157426017,
  "node_interface_network_tx_bytes": 355.385094878293,
  "node_interface_network_tx_dropped": 0,
  "node_interface_network_tx_errors": 0,
  "node_interface_network_tx_packets": 3.9997714100370625
}
```


Typ: Pod


```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "pod_cpu_utilization"
        },
```

```
        {
          "Unit": "Percent",
          "Name": "pod_memory_utilization"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "pod_network_rx_bytes"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "pod_network_tx_bytes"
        },
        {
          "Unit": "Percent",
          "Name": "pod_cpu_utilization_over_pod_limit"
        },
        {
          "Unit": "Percent",
          "Name": "pod_memory_utilization_over_pod_limit"
        }
      ],
      "Dimensions": [
        [
          "PodName",
          "Namespace",
          "ClusterName"
        ],
        [
          "Service",
          "Namespace",
          "ClusterName"
        ],
        [
          "Namespace",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    },
    {
      "Metrics": [
```

```
        {
          "Unit": "Percent",
          "Name": "pod_cpu_reserved_capacity"
        },
        {
          "Unit": "Percent",
          "Name": "pod_memory_reserved_capacity"
        }
      ],
      "Dimensions": [
        [
          "PodName",
          "Namespace",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    },
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "pod_number_of_container_restarts"
        }
      ],
      "Dimensions": [
        [
          "PodName",
          "Namespace",
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
```

```
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "pod",
    "calculated"
  ],
  "Timestamp": "1567097351092",
  "Type": "Pod",
  "Version": "0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  },
  "pod_cpu_limit": 200,
  "pod_cpu_request": 200,
  "pod_cpu_reserved_capacity": 5,
  "pod_cpu_usage_system": 1.4504841104992765,
  "pod_cpu_usage_total": 5.817016867430125,
  "pod_cpu_usage_user": 1.1281543081661038,
  "pod_cpu_utilization": 0.14542542168575312,
  "pod_cpu_utilization_over_pod_limit": 2.9085084337150624,
  "pod_memory_cache": 8192,
  "pod_memory_failcnt": 0,
  "pod_memory_hierarchical_pgfault": 0,
  "pod_memory_hierarchical_pgmajfault": 0,
  "pod_memory_limit": 104857600,
  "pod_memory_mapped_file": 0,
  "pod_memory_max_usage": 25268224,
  "pod_memory_pgfault": 0,
  "pod_memory_pgmajfault": 0,
  "pod_memory_request": 104857600,
```

```
  "pod_memory_reserved_capacity": 0.6307275170893897,
  "pod_memory_rss": 22777856,
  "pod_memory_swap": 0,
  "pod_memory_usage": 25141248,
  "pod_memory_utilization": 0.10988455961791709,
  "pod_memory_utilization_over_pod_limit": 17.421875,
  "pod_memory_working_set": 18268160,
  "pod_network_rx_bytes": 9880.697124714186,
  "pod_network_rx_dropped": 0,
  "pod_network_rx_errors": 0,
  "pod_network_rx_packets": 107.80005532263283,
  "pod_network_total_bytes": 10158.829201483635,
  "pod_network_tx_bytes": 278.13207676944796,
  "pod_network_tx_dropped": 0,
  "pod_network_tx_errors": 0,
  "pod_network_tx_packets": 1.146027574644318,
  "pod_number_of_container_restarts": 0,
  "pod_number_of_containers": 1,
  "pod_number_of_running_containers": 1,
  "pod_status": "Running"
}
```

Typ: PodNet

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567097351092",
  "Type": "PodNet",
  "Version": "0",
  "interface": "eth0",
  "kubernetes": {
```

```
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  },
  "pod_interface_network_rx_bytes": 9880.697124714186,
  "pod_interface_network_rx_dropped": 0,
  "pod_interface_network_rx_errors": 0,
  "pod_interface_network_rx_packets": 107.80005532263283,
  "pod_interface_network_total_bytes": 10158.829201483635,
  "pod_interface_network_tx_bytes": 278.13207676944796,
  "pod_interface_network_tx_dropped": 0,
  "pod_interface_network_tx_errors": 0,
  "pod_interface_network_tx_packets": 1.146027574644318
}
```

Typ: Container

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-NodeGroup-
sample",
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "pod",
    "calculated"
```

```
  ],
  "Timestamp": "1567097399912",
  "Type": "Container",
  "Version": "0",
  "container_cpu_limit": 200,
  "container_cpu_request": 200,
  "container_cpu_usage_system": 1.87958283771964,
  "container_cpu_usage_total": 6.159993652997942,
  "container_cpu_usage_user": 1.6707403001952357,
  "container_cpu_utilization": 0.15399984132494854,
  "container_memory_cache": 8192,
  "container_memory_failcnt": 0,
  "container_memory_hierarchical_pgfault": 0,
  "container_memory_hierarchical_pgmajfault": 0,
  "container_memory_limit": 104857600,
  "container_memory_mapped_file": 0,
  "container_memory_max_usage": 24580096,
  "container_memory_pgfault": 0,
  "container_memory_pgmajfault": 0,
  "container_memory_request": 104857600,
  "container_memory_rss": 22736896,
  "container_memory_swap": 0,
  "container_memory_usage": 24453120,
  "container_memory_utilization": 0.10574541028701798,
  "container_memory_working_set": 17580032,
  "container_status": "Running",
  "kubernetes": {
    "container_name": "cloudwatch-agent",
    "docker": {
      "container_id":
"8967b6b37da239dfad197c9fdea3e5dfd35a8a759ec86e2e4c3f7b401e232706"
    },
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
```

```
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  },
  "number_of_container_restarts": 0
}
```

Typ: ContainerFS

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "EBSVolumeId": "aws://us-west-2b/vol-0a53108976d4a2fda",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567097399912",
  "Type": "ContainerFS",
  "Version": "0",

  "device": "/dev/nvme0n1p1",
  "fstype": "vfs",
  "kubernetes": {
    "container_name": "cloudwatch-agent",
    "docker": {
      "container_id":
 "8967b6b37da239dfad197c9fdea3e5dfd35a8a759ec86e2e4c3f7b401e232706"
    },
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
```

```
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  }
}
```

Typ: Cluster

```
{
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "cluster_node_count"
        },
        {
          "Unit": "Count",
          "Name": "cluster_failed_node_count"
        }
      ],
      "Dimensions": [
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "Sources": [
    "apiserver"
  ],
  "Timestamp": "1567097534160",
  "Type": "Cluster",
  "Version": "0",
  "cluster_failed_node_count": 0,
  "cluster_node_count": 3
```

```
}
```

Typ: ClusterService

```
{
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "service_number_of_running_pods"
        }
      ],
      "Dimensions": [
        [
          "Service",
          "Namespace",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "Namespace": "amazon-cloudwatch",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "apiserver"
  ],
  "Timestamp": "1567097534160",
  "Type": "ClusterService",
  "Version": "0",
  "kubernetes": {
    "namespace_name": "amazon-cloudwatch",
    "service_name": "cloudwatch-agent-statsd"
  },
  "service_number_of_running_pods": 1
}
```

Typ: ClusterNamespace

```
{
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "namespace_number_of_running_pods"
        }
      ],
      "Dimensions": [
        [
          "Namespace",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "Namespace": "amazon-cloudwatch",
  "Sources": [
    "apiserver"
  ],
  "Timestamp": "1567097594160",
  "Type": "ClusterNamespace",
  "Version": "0",
  "kubernetes": {
    "namespace_name": "amazon-cloudwatch"
  },
  "namespace_number_of_running_pods": 7
}
```

Relevante Felder in Performance-Protokollereignissen für Amazon EKS und Kubernetes

Für Amazon EKS und Kubernetes gibt der containerisierte CloudWatch Agent Daten als Leistungsprotokollereignisse aus. Dies ermöglicht die Aufnahme und Speicherung von Daten CloudWatch mit hoher Kardinalität. CloudWatch verwendet die Daten in den

Leistungsprotokollereignissen, um aggregierte CloudWatch Metriken auf Cluster-, Knoten- und Pod-Ebene zu erstellen, ohne dass detaillierte Details verloren gehen müssen.

In der folgenden Tabelle werden die Felder in diesen Performance-Protokollereignissen aufgelistet, die für die Sammlung von Container Insights-Metrikdaten relevant sind. Sie können CloudWatch Logs Insights verwenden, um jedes dieser Felder abzufragen, um Daten zu sammeln oder Probleme zu untersuchen. Weitere Informationen finden Sie unter Analysieren von Protokolldaten mit CloudWatch Logs Insights.

Typ	Protokollfeld	Quelle	Formel oder Hinweise
Pod	`pod_cpu_utilization`	Berechnet	Formel: `pod_cpu_usage_total / node_cpu_limit`
Pod	`pod_cpu_usage_total` `pod_cpu_usage_total` wird in Millicore gemeldet.	cadvisor	
Pod	`pod_cpu_limit`	Berechnet	Formel: `sum(container_cpu_limit)` `sum(container_cpu_limit)` beinhaltet bereits abgeschlossene Pods. Dieses Feld erscheint nur

Typ	Protokollfeld	Quelle	Formel oder Hinweise
			im Protokoll ereignis, wenn für alle Container im Pod ein CPU-Limit definiert ist. Dies schließt Init-Container ein.
Pod	`pod_cpu_request`	Berechnet	Formel: `sum(container_cpu_request)` Es ist nicht garantiert, dass `container_cpu_request` eingestellt wird. Nur diejenigen, die eingestellt sind, sind in der Summe enthalten.
Pod	`pod_cpu_utilization_over_pod_limit`	Berechnet	Formel: `pod_cpu_usage_total / pod_cpu_limit`

Typ	Protokollfeld	Quelle	Formel oder Hinweise
Pod	`pod_cpu_reserved_capacity`	Berechnet	Formel: `pod_cpu_request / node_cpu_limit`
Pod	`pod_memory_utilization`	Berechnet	Formel: `pod_memory_working_set / node_memory_limit` Dies ist der Prozentsatz der Pod-Speicherhernutzung bezüglich der Knotenspeicherbegrenzung.
Pod	`pod_memory_working_set`	cadvisor	

Typ	Protokollfeld	Quelle	Formel oder Hinweise
Pod	`pod_memory_limit`	Berechnet	Formel: `sum(container_memory_limit)` Dieses Feld erscheint nur im Protokollereignis, wenn für alle Container im Pod ein Arbeitsspeicherlimit definiert ist. Dies schließt Init-Container ein.

Typ	Protokollfeld	Quelle	Formel oder Hinweise
Pod	`pod_memory_request`	Berechnet	Formel: `sum(container_memory_request)` Es ist nicht garantiert, dass `container_memory_request` eingestellt wird. Nur diejenigen, die eingestellt sind, sind in der Summe enthalten.

Typ	Protokollfeld	Quelle	Formel oder Hinweise
Pod	`pod_memory_utilization_over_pod_limit`	Berechnet	Formel: `pod_memory_working_set / pod_memory_limit` Dieses Feld erscheint nur im Protokollereignis, wenn für alle Container im Pod ein Arbeitsspeicherlimit definiert ist. Dies schließt Init-Container ein.
Pod	`pod_memory_reserved_capacity`	Berechnet	Formel: `pod_memory_request / node_memory_limit`

Typ	Protokollfeld	Quelle	Formel oder Hinweise
Pod	pod_network_tx_bytes	Berechnet	Formel: sum(pod_interface_network_tx_bytes) Diese Daten sind für alle Netzwerkschnittstellen pro Pod verfügbar. Der CloudWatch Agent berechnet die Gesamtsumme und fügt Regeln für die Extraktion von Metriken hinzu.
Pod	pod_network_rx_bytes	Berechnet	Formel: sum(pod_interface_network_rx_bytes)
Pod	pod_network_total_bytes	Berechnet	Formel: pod_network_rx_bytes + pod_network_tx_bytes

Typ	Protokollfeld	Quelle	Formel oder Hinweise
PodNet	`pod_interface_network_rx_bytes`	cadvisor	Diese Daten sind rx-Bytes des Netzwerks pro Sekunde einer Pod-Netzwerkschnittstelle.
PodNet	`pod_interface_network_tx_bytes`	cadvisor	Diese Daten sind tx-Bytes des Netzwerks pro Sekunde einer Pod-Netzwerkschnittstelle.
Container	`container_cpu_usage_total`	cadvisor	
Container	`container_cpu_limit`	cadvisor	Es wird nicht garantiert, dass es festgelegt wird. Es wird nicht ausgegeben, wenn es nicht festgelegt ist.

Typ	Protokollfeld	Quelle	Formel oder Hinweise
Container	`container_cpu_request`	cadvisor	Es wird nicht garantiert, dass es festgeleg t wird. Es wird nicht ausgegeben, wenn es nicht festgelegt ist.
Container	`container_memory_working_set`	cadvisor	
Container	`container_memory_limit`	Pod	Es wird nicht garantiert, dass es festgeleg t wird. Es wird nicht ausgegeben, wenn es nicht festgelegt ist.
Container	`container_memory_request`	Pod	Es wird nicht garantiert, dass es festgeleg t wird. Es wird nicht ausgegeben, wenn es nicht festgelegt ist.

Typ	Protokollfeld	Quelle	Formel oder Hinweise
Knoten	node_cpu_utilization	Berechnet	Formel: node_cpu_usage_total / node_cpu_limit
Knoten	node_cpu_usage_total	cadvisor	
Knoten	node_cpu_limit	/proc	
Knoten	node_cpu_request	Berechnet	Formel: sum(pod_cpu_request) node_cpu_request schließt bei Cronjobs auch Anfragen von abgeschlossenen Pods ein. Dies kann zu einem hohen Wert für node_cpu_reserved_capacity führen.

Typ	Protokollfeld	Quelle	Formel oder Hinweise
Knoten	node_cpu_reserved_capacity	Berechnet	Formel: node_cpu_request / node_cpu_limit
Knoten	node_memory_utilization	Berechnet	Formel: node_memory_working_set / node_memory_limit
Knoten	node_memory_working_set	cadvisor	
Knoten	node_memory_limit	/proc	
Knoten	node_memory_request	Berechnet	Formel: sum(pod_memory_request)
Knoten	node_memory_reserved_capacity	Berechnet	Formel: node_memory_request / node_memory_limit
Knoten	node_network_rx_bytes	Berechnet	Formel: sum(node_interface_network_rx_bytes)

Typ	Protokollfeld	Quelle	Formel oder Hinweise
Knoten	`node_network_tx_bytes`	Berechnet	Formel: `sum(node_interface_network_tx_bytes)`
Knoten	`node_network_total_bytes`	Berechnet	Formel: `node_network_rx_bytes + node_network_tx_bytes`
Knoten	`node_number_of_running_pods`	Pod-Liste	
Knoten	`node_number_of_running_containers`	Pod-Liste	
NodeNet	`node_interface_network_rx_bytes`	cadvisor	Diese Daten sind Netzwerk-rx-Bytes pro Sekunde der Netzwerkschnittstelle eines Workerknotens.

Typ	Protokollfeld	Quelle	Formel oder Hinweise
NodeNet	`node_interface_network_tx_bytes`	cadvisor	Diese Daten sind Netzwerk-tx-Bytes pro Sekunde der Netzwerkschnittstelle eines Workerknotens.
NodeFS	`node_filesystem_capacity`	cadvisor	
NodeFS	`node_filesystem_usage`	cadvisor	
NodeFS	`node_filesystem_utilization`	Berechnet	Formel: `node_filesystem_usage / node_filesystem_capacity` Diese Daten sind pro Gerätename verfügbar.
Cluster	`cluster_failed_node_count`	API-Server	
Cluster	`cluster_node_count`	API-Server	
Service	`service_number_of_running_pods`	API-Server	

Typ	Protokollfeld	Quelle	Formel oder Hinweise
Namespace	namespace_number_of_running_pods	API-Server	

Beispiele für Metrikberechnungen

Dieser Abschnitt enthält Beispiele zur Veranschaulichung, wie einige der Werte in der vorangegangenen Tabelle berechnet werden.

Angenommen, Sie besitzen einen Cluster im folgenden Zustand.

```
Node1
   node_cpu_limit = 4
   node_cpu_usage_total = 3

   Pod1
     pod_cpu_usage_total = 2

     Container1
        container_cpu_limit = 1
        container_cpu_request = 1
        container_cpu_usage_total = 0.8

     Container2
        container_cpu_limit = null
        container_cpu_request = null
        container_cpu_usage_total = 1.2

   Pod2
     pod_cpu_usage_total = 0.4

     Container3
        container_cpu_limit = 1
        container_cpu_request = 0.5
        container_cpu_usage_total = 0.4

Node2
   node_cpu_limit = 8
   node_cpu_usage_total = 1.5
```

```
  Pod3
    pod_cpu_usage_total = 1

    Container4
       container_cpu_limit = 2
       container_cpu_request = 2
       container_cpu_usage_total = 1
```

Die folgende Tabelle zeigt, wie Pod-CPU-Metriken anhand dieser Daten berechnet werden.

Metrik	Formel	Pod1	Pod2	Pod3
`pod_cpu_utilization`	`pod_cpu_usage_total / node_cpu_limit`	2/4 = 50 %	0,4/4 = 10 %	1/8 = 12,5 %
`pod_cpu_utilization_over_pod_limit`	`pod_cpu_usage_total / sum(container_cpu_limit)`	Nicht zutreffend, da kein CPU-Limit für `Container2` nicht definiert ist	0,4/1 = 40 %	1/2 = 50 %
`pod_cpu_reserved_capacity`	`sum(container_cpu_request) / node_cpu_limit`	(1 + 0)/4 = 25 %	0,5/4 = 12,5 %	2/8 = 25 %

Die folgende Tabelle zeigt, wie Knoten-CPU-Metriken anhand dieser Daten berechnet werden.

Metrik	Formel	Knoten1	Knoten2
`node_cpu_utilization`	`node_cpu_usage_total / node_cpu_limit`	3/4 = 75 %	1,5/8 = 18,75 %
`node_cpu_reserved_capacity`	`sum(pod_cpu_request) / node_cpu_limit`	1,5/4 = 37,5 %	2/8 = 25 %

Überwachung von Container Insights Prometheus-Metriken

CloudWatch Die Überwachung von Container Insights für Prometheus automatisiert die Erkennung von Prometheus-Metriken aus containerisierten Systemen und Workloads. Prometheus ist ein Open-Source-Toolkit zur Überwachung und Benachrichtigung für Systeme. Weitere Informationen finden Sie unter Was ist Prometheus? in der Prometheus-Dokumentation.

Das Erkennen von Prometheus-Metriken wird für Amazon Elastic Container Service, Amazon Elastic Kubernetes Service und Kubernetes-Cluster unterstützt, die auf Amazon-EC2-Instances ausgeführt werden. Die Prometheus-Zähler-, Messinstrument- und Zusammenfassungsmetriktypen werden erfasst. Unterstützung für Histogrammmetriken ist für eine kommende Version geplant.

Für Amazon-ECS- und Amazon-EKS-Cluster werden sowohl die Starttypen EC2 als auch Fargate unterstützt. Container Insights erfasst automatisch Metriken aus mehreren Workloads. Sie können sie so konfigurieren, dass Metriken aus jeder Workload erfasst werden.

Sie können Prometheus als Open Source- und Open-Standard-Methode für die Aufnahme benutzerdefinierter Metriken einsetzen. CloudWatch Der CloudWatch Agent mit Prometheus-Unterstützung erkennt und sammelt Prometheus-Metriken, um Leistungseinbußen und Ausfälle von Anwendungen schneller zu überwachen, Fehler zu beheben und Warnmeldungen zu geben. Dies reduziert auch die Anzahl der Tools, die zur Verbesserung der Überwachung erforderlich sind.

Die Unterstützung pay-per-use von Container Insights Prometheus umfasst Metriken und Protokolle, einschließlich der Erfassung, Speicherung und Analyse. Weitere Informationen finden Sie unter CloudWatch Amazon-Preise.

Vorgefertigte Dashboards für einige Workloads

Die Container-Insights-Prometheus-Lösung enthält vorgefertigte Dashboards für die in diesem Abschnitt aufgeführten beliebten Workloads. Beispielkonfigurationen für diese Workloads finden Sie unter (Optional) Einrichten von containerisierten Beispiel-Amazon-ECS-Workloads für Prometheus-Metrik-Tests und (Optional) Einrichten von containerisierten Beispiel-Amazon-EKS-Workloads für Prometheus-Metrik-Tests.

Sie können Container Insights auch so konfigurieren, dass Prometheus-Metriken von anderen containerisierten Services und Anwendungen erfasst werden, indem Sie die Agentkonfigurationsdatei bearbeiten.

Workloads mit vordefinierten Dashbords für Amazon-EKS-Cluster und Kubernetes-Cluster, die auf Amazon-EC2-Instances ausgeführt werden:

- AWS App Mesh
- NGINX
- Memcached
- Java/JMX
- HAProxy

Workloads mit vordefinierten Dashbords für Amazon-ECS-Cluster:

- AWS App Mesh
- Java/JMX
- NGINX
- NGINX Plus

Einrichten und Konfigurieren der Prometheus-Metrikensammlung in Amazon-ECS-Clustern

Um Prometheus-Metriken aus Amazon ECS-Clustern zu sammeln, können Sie den CloudWatch Agenten als Collector oder die AWS Distro for Collector verwenden. OpenTelemetry Informationen zur Verwendung von AWS Distro for OpenTelemetry Collector finden Sie unter https://aws-otel.github.io/docs/getting-started/container-insights/ecs-prometheus.

In den folgenden Abschnitten wird erklärt, wie der CloudWatch Agent als Collector zum Abrufen von Prometheus-Metriken verwendet wird. Sie installieren den CloudWatch Agenten mit Prometheus-Überwachung auf Clustern, auf denen Amazon ECS ausgeführt wird, und Sie können den Agenten optional so konfigurieren, dass er zusätzliche Ziele scannt. Diese Abschnitte enthalten auch optionale Tutorials zum Einrichten von Beispiel-Workloads zum Testen mit der Prometheus-Überwachung.

Container Insights auf Amazon ECS unterstützt die folgenden Kombinationen von Starttyp und Netzwerkmodus für Prometheus Metriken:

Amazon-ECS-Starttyp	Unterstützte Netzwerkmodi
EC2 (Linux)	bridge, Host und awsvpc
Fargate	awsvpc

Anforderungen an VPC-Sicherheitsgruppen

Die Eingangsregeln der Sicherheitsgruppen für die Prometheus-Workloads müssen die Prometheus-Ports für den CloudWatch Agenten öffnen, damit er die Prometheus-Metriken über die private IP scrapen kann.

Die Ausgangsregeln der Sicherheitsgruppe für den CloudWatch Agenten müssen es dem CloudWatch Agenten ermöglichen, über eine private IP eine Verbindung zum Port der Prometheus-Workloads herzustellen.

Themen

Installieren Sie den CloudWatch Agenten mit der Erfassung von Prometheus-Metriken auf Amazon ECS-Clustern

In diesem Abschnitt wird erklärt, wie Sie den CloudWatch Agenten mit Prometheus-Überwachung in einem Cluster einrichten, auf dem Amazon ECS ausgeführt wird. Danach scrapt und importiert der Agent automatisch Metriken für die folgenden Workloads, die in diesem Cluster ausgeführt werden.

- AWS App Mesh
- Java/JMX

Sie können den Agenten auch so konfigurieren, dass er Metriken aus weiteren Prometheus-Workloads und -Quellen importiert.

Einrichten von IAM-Rollen

Für die Aufgabendefinition des Agenten benötigen Sie zwei IAM-Rollen. CloudWatch Wenn Sie **`CreateIAMRoles=True`** im AWS CloudFormation Stack angeben, dass Container Insights diese Rollen für Sie erstellen soll, werden die Rollen mit den richtigen Berechtigungen erstellt. Wenn Sie sie selbst erstellen oder vorhandene Rollen verwenden möchten, sind die folgenden Rollen und Berechtigungen erforderlich.

- CloudWatch ECS-Aufgabenrolle für Agenten — Der CloudWatch Agent-Container verwendet diese Rolle. Sie muss die CloudWatchAgentServerPolicyRichtlinie und eine vom Kunden verwaltete Richtlinie enthalten, die die folgenden schreibgeschützten Berechtigungen enthält:
  - `ec2:DescribeInstances`
  - `ecs:ListTasks`
  - `ecs:ListServices`
  - `ecs:DescribeContainerInstances`
  - `ecs:DescribeServices`
  - `ecs:DescribeTasks`
  - `ecs:DescribeTaskDefinition`
- CloudWatch Rolle zur Ausführung von ECS-Aufgaben für Agenten — Dies ist die Rolle, die Amazon ECS benötigt, um Ihre Container zu starten und auszuführen. Stellen Sie sicher, dass Ihrer Aufgabenausführungsrolle die AmazonSSM -ReadOnlyAccess, AmazonECS - und CloudWatchAgentServerPolicyRichtlinien TaskExecutionRolePolicy zugeordnet sind. Wenn Sie sensiblere Daten zur Verwendung durch Amazon ECS speichern möchten, finden Sie weitere Informationen unter Angabe sensibler Daten.

Installieren Sie den CloudWatch Agenten mit Prometheus-Überwachung mithilfe von AWS CloudFormation

Sie verwenden AWS CloudFormation , um den CloudWatch Agenten mit Prometheus-Überwachung für Amazon ECS-Cluster zu installieren. Die folgende Liste zeigt die Parameter, die Sie in der AWS CloudFormation -Vorlage verwenden werden.

- ECS ClusterName — Gibt den Amazon ECS-Zielcluster an.
- CreateIamRollen – Geben Sie **`True`** an, um neue Rollen für die Amazon-ECS-Aufgabenrolle und die Amazon-ECS-Aufgabenausführungsrolle zu erstellen. Geben Sie **`False`** an, um vorhandene Rollen wiederzuverwenden.
- TaskRoleName— Wenn Sie CreateIamRoles angegeben **`True`** haben, gibt dies den Namen an, der für die neue Amazon ECS-Aufgabenrolle verwendet werden soll. Wenn Sie **`False`** für CreateIAMRoles angegeben haben, gibt dies den Namen an, der für die neue Amazon-ECS-Aufgabenrolle verwendet werden soll.
- ExecutionRoleName— Wenn Sie CreateIamRoles angegeben **`True`** haben, gibt dies den Namen an, der für die neue Amazon ECS-Aufgabenausführungsrolle verwendet werden soll. Wenn Sie

`False` für CreateIAMRoles angegeben haben, gibt dies den Namen an, der für die neue Amazon-ECS-Aufgabenausführungsrolle verwendet werden soll.

- ECS NetworkMode — Wenn Sie den EC2-Starttyp verwenden, geben Sie hier den Netzwerkmodus an. Es muss entweder **`bridge`** oder **`host`** sein.
- ECS LaunchType — Geben Sie entweder **`fargate`** oder **`EC2`** an.
- SecurityGroupID — Falls ECS vorhanden NetworkMode ist**`awsvpc`**, geben Sie hier die Sicherheitsgruppen-ID an.
- SubnetID — Wenn das ECS vorhanden NetworkMode ist**`awsvpc`**, geben Sie hier die Subnetz-ID an.

Befehlsbeispiele

Dieser Abschnitt enthält AWS CloudFormation Beispielbefehle zur Installation von Container Insights mit Prometheus-Überwachung in verschiedenen Szenarien.

AWS CloudFormation Stack für einen Amazon ECS-Cluster im Bridge-Netzwerkmodus erstellen

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_NETWORK_MODE=bridge
export CREATE_IAM_ROLES=True
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-bridge-host.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSNetworkMode,ParameterValue=${ECS_NETWORK_MODE} \
                 ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
```

```
    --profile ${AWS_PROFILE}
```

AWS CloudFormation Stack für einen Amazon ECS-Cluster im Host-Netzwerkmodus erstellen

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_NETWORK_MODE=host
export CREATE_IAM_ROLES=True
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name


curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-bridge-host.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSNetworkMode,ParameterValue=${ECS_NETWORK_MODE} \
                 ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

Erstellen Sie einen AWS CloudFormation Stack für einen Amazon ECS-Cluster im awsvpc-Netzwerkmodus

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_LAUNCH_TYPE=EC2
export CREATE_IAM_ROLES=True
export ECS_CLUSTER_SECURITY_GROUP=your_security_group_eg_sg-xxxxxxxxxx
export ECS_CLUSTER_SUBNET=your_subnet_eg_subnet-xxxxxxxxxx
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name
```

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-awsvpc.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-${ECS_LAUNCH_TYPE}-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSLaunchType,ParameterValue=${ECS_LAUNCH_TYPE} \
                 ParameterKey=SecurityGroupID,ParameterValue=
${ECS_CLUSTER_SECURITY_GROUP} \
                 ParameterKey=SubnetID,ParameterValue=${ECS_CLUSTER_SUBNET} \
                 ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

Erstellen Sie einen AWS CloudFormation Stack für einen Fargate-Cluster im awsvpc-Netzwerkmodus

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_LAUNCH_TYPE=FARGATE
export CREATE_IAM_ROLES=True
export ECS_CLUSTER_SECURITY_GROUP=your_security_group_eg_sg-xxxxxxxxxx
export ECS_CLUSTER_SUBNET=your_subnet_eg_subnet-xxxxxxxxxx
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-awsvpc.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-${ECS_LAUNCH_TYPE}-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSLaunchType,ParameterValue=${ECS_LAUNCH_TYPE} \
```

```
                ParameterKey=SecurityGroupID,ParameterValue=
${ECS_CLUSTER_SECURITY_GROUP} \
                ParameterKey=SubnetID,ParameterValue=${ECS_CLUSTER_SUBNET} \
                ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

AWS Ressourcen, die durch den Stack erstellt wurden AWS CloudFormation

In der folgenden Tabelle sind die AWS Ressourcen aufgeführt, die erstellt werden, wenn Sie AWS CloudFormation Container Insights mit Prometheus-Überwachung auf einem Amazon ECS-Cluster einrichten.

Ressourcentyp	Ressourcenname	Kommentare
AWS::SSM::Parameter	AmazonCloudWatch-*CW -$ ECS_CLUSTER_NAME -$ AgentConfig ECS_LAUNCH_TYPE -$ ECS_NETWORK_MODE*	Dies ist der CloudWatch Agent mit der standardmäßigen metrischen Formatdefinition für App Mesh und Java/JMX.
AWS::SSM::Parameter	AmazonCloudWatch-*Prometheu sConfigName -$ ECS_CLUSTER_NAME -$ ECS_LAUNCH_TYPE -$ ECS_NETWORK_MODE*	Dies ist die Prometheus-Scraping-Konfiguration.
AWS::IAM::Role	$ECS_TASK_ROLE_NAME.	Die Amazon-ECS-Aufgabenrolle. Dies wird nur erstellt, wenn Sie **`True`**für `CREATE_IAM_ROLES` angegeben haben.
AWS::IAM::Role	${ECS_EXECUTION_ROLE_NAME}	IAM-Rolle für die Amazon-ECS-Aufgabenausführung Dies wird nur erstellt, wenn Sie **`True`**für `CREATE_IAM_ROLES` angegeben haben.

Ressourcentyp	Ressourcenname	Kommentare
AWS::ECS::TaskDefinition	cwagent-prometheus-$*ECS_CLUSTER_NAME* -$*ECS_LAUNCH_TYPE* -$*ECS_NETWORK_MODE*	
AWS::ECS::Service	cwagent-prometheus-replica-service-$ECS_LAUNCH_TYPE -$ ECS_NETWORK_MODE	

Löschen des AWS CloudFormation Stacks für den CloudWatch Agenten mit Prometheus-Überwachung

Um den CloudWatch Agenten aus einem Amazon ECS-Cluster zu löschen, geben Sie diese Befehle ein.

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export CLOUDFORMATION_STACK_NAME=your_cloudformation_stack_name

aws cloudformation delete-stack \
--stack-name ${CLOUDFORMATION_STACK_NAME} \
--region ${AWS_DEFAULT_REGION} \
--profile ${AWS_PROFILE}
```

Scraping zusätzlicher Prometheus-Quellen und Importieren dieser Metriken

Der CloudWatch Agent mit Prometheus-Überwachung benötigt zwei Konfigurationen, um die Prometheus-Metriken zu erfassen. Er folgt der standardmäßigen Prometheus-Konfiguration, wie in <scrape_config> in der Prometheus-Dokumentation erläutert. Die andere ist für die Agentenkonfiguration vorgesehen. CloudWatch

Für Amazon-ECS-Cluster werden die Konfigurationen durch die Secrets in der Amazon-ECS-Aufgabendefinition in den Parameter Store von AWS Systems Manager integriert:

- Das Secret `PROMETHEUS_CONFIG_CONTENT` ist für die Scrape-Konfiguration von Prometheus.
- Das Geheimnis bezieht `CW_CONFIG_CONTENT` sich auf die CloudWatch Agentenkonfiguration.

Um zusätzliche Prometheus-Metrikquellen zu scrapen und diese Metriken zu importieren CloudWatch, ändern Sie sowohl die Prometheus-Scrape-Konfiguration als auch die Agentenkonfiguration und stellen dann den CloudWatch Agenten mit der aktualisierten Konfiguration erneut bereit.

Anforderungen an VPC-Sicherheitsgruppen

Die Eingangsregeln der Sicherheitsgruppen für die Prometheus-Workloads müssen die Prometheus-Ports für den CloudWatch Agenten öffnen, damit er die Prometheus-Metriken über die private IP scrapen kann.

Die Ausgangsregeln der Sicherheitsgruppe für den CloudWatch Agenten müssen es dem CloudWatch Agenten ermöglichen, über eine private IP eine Verbindung zum Port der Prometheus-Workloads herzustellen.

Prometheus-Scrape-Konfiguration

Der CloudWatch Agent unterstützt die standardmäßigen Prometheus-Scrape-Konfigurationen, wiehttps://prometheus.io/docs/prometheus/latest/configuration/configuration/#scrape_config <scrape_config>in der Prometheus-Dokumentation dokumentiert. Sie können diesen Abschnitt bearbeiten, um die Konfigurationen zu aktualisieren, die sich bereits in dieser Datei befinden, und zusätzliche Prometheus-Scraping-Ziele hinzufügen. Standardmäßig enthält die Beispielkonfigurationsdatei die folgenden globalen Konfigurationszeilen:

```
global:
  scrape_interval: 1m
  scrape_timeout: 10s
```

- scrape_interval – Definiert, wie oft das Scraping von zielen durchgeführt werden soll.
- scrape_timeout – Definiert, wie lange gewartet werden soll, bis für eine Scrape-Anforderung eine Zeitüberschreitung eintritt.

Sie können auch verschiedene Werte für diese Einstellungen auf Auftragsebene definieren, um die globalen Konfigurationen zu überschreiben.

Prometheus-Scraping-Aufträge

Für die YAML-Dateien des CloudWatch Agenten sind bereits einige Standard-Scraping-Jobs konfiguriert. In den YAML-Dateien für Amazon ECS wie `cwagent-ecs-prometheus-metric-`

`for-bridge-host.yaml` werden beispielsweise die Standard-Scraping-Aufträge im Abschnitt `ecs_service_discovery` konfiguriert.

```
"ecs_service_discovery": {
                  "sd_frequency": "1m",
                  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
                  "docker_label": {
                  },
                  "task_definition_list": [
                    {
                      "sd_job_name": "ecs-appmesh-colors",
                      "sd_metrics_ports": "9901",
                      "sd_task_definition_arn_pattern": ".*:task-definition\/.*-
ColorTeller-(white):[0-9]+",
                      "sd_metrics_path": "/stats/prometheus"
                    },
                    {
                      "sd_job_name": "ecs-appmesh-gateway",
                      "sd_metrics_ports": "9901",
                      "sd_task_definition_arn_pattern": ".*:task-definition/.*-
ColorGateway:[0-9]+",
                      "sd_metrics_path": "/stats/prometheus"
                    }
                  ]
                }
```

Jedes dieser Standardziele wird gelöscht, und die Metriken werden im eingebetteten Metrikformat CloudWatch an Protokollereignisse gesendet. Weitere Informationen finden Sie unter Einbetten von Metriken in Protokollen.

Protokollereignisse von Amazon-ECS-Clustern werden in der Protokollgruppe /aws/ecs/containerinsights/***cluster_name***/prometheus gespeichert.

Jeder Scraping-Auftrag ist in einem anderen Protokoll-Stream in dieser Protokollgruppe enthalten.

Um ein neues Scraping-Ziel hinzuzufügen, fügen Sie im Abschnitt `task_definition_list` unter dem Abschnitt `ecs_service_discovery` der YAML-Datei einen neuen Eintrag hinzu und starten den Agenten neu. Ein Beispiel für diesen Prozess finden Sie unter Tutorial zum Hinzufügen eines neuen Prometheus-Scrape-Ziels: Prometheus-API-Server-Metriken.

CloudWatch Agentenkonfiguration für Prometheus

Die CloudWatch Agentenkonfigurationsdatei enthält einen `prometheus` Abschnitt `metrics_collected` für die Prometheus-Scraping-Konfiguration. Es sind folgende Konfigurationsoptionen enthalten:

- Clustername – Gibt den Clusternamen an, der als Bezeichnung im Protokollereignis hinzugefügt werden soll. Dies ist ein optionales Feld. Wenn Sie es weglassen, kann der Agent den Amazon-ECS-Clusternamen erkennen.
- log_group_name – Gibt den Namen der Protokollgruppe für die Prometheus-Scrape-Metriken an. Dies ist ein optionales Feld. *Wenn Sie es weglassen, wird* ***/aws/ecs/containerinsights/ cluster_name /prometheus für Protokolle von Amazon ECS-Clustern CloudWatch*** *verwendet.*
- prometheus_config_path – gibt den Pfad der Prometheus-Scrape-Konfigurationsdatei an. Wenn der Wert dieses Felds mit `env:` beginnt, wird der Inhalt der Prometheus-Scrape-Konfigurationsdatei aus der Umgebungsvariablen des Containers abgerufen. Ändern Sie dieses Feld nicht.
- ecs_service_discovery – ist der Abschnitt zum Angeben der Konfigurationen der Ziel-Auto-Discovery-Funktionen von Amazon ECS Prometheus. Zur Ermittlung der Prometheus-Ziele werden zwei Modi unterstützt: Ermittlung basierend auf der Docker-Bezeichnung des Containers oder Ermittlung basierend auf dem regulären ARN-Ausdruck der Amazon-ECS-Aufgabendefinition. Sie können die beiden Modi zusammen verwenden und der CloudWatch Agent dedupliziert die erkannten Ziele auf der Grundlage von: {private_ip}: {port}/{metrics_path}.

  Der Abschnitt `ecs_service_discovery` kann die folgenden Felder enthalten:

  - `sd_frequency` ist die Häufigkeit, mit der die Prometheus-Exporteure entdeckt werden. Geben Sie eine Zahl und ein Einheitensuffix an. Zum Beispiel `1m` für einmal pro Minute oder `30s` für einmal pro 30 Sekunden. Gültige Einheitensuffixe sind `ns`, `us`, `ms`, `s`, `m` und `h`.

    Dies ist ein optionales Feld. Der Standardwert ist 60 Sekunden (1 Minute).
  - `sd_target_cluster` ist der Name des Amazon-ECS-Ziel-Clusters für die automatische Erkennung. Dies ist ein optionales Feld. Der Standard ist der Name des Amazon ECS-Clusters, auf dem der CloudWatch Agent installiert ist.
  - `sd_cluster_region` ist die Region des Amazon-ECS-Ziel-Clusters. Dies ist ein optionales Feld. Die Standardeinstellung ist die Region des Amazon ECS-Clusters, in der der CloudWatch Agent installiert ist.

- `sd_result_file` ist der Pfad der YAML-Datei für die Prometheus Zielergebnisse. Die Prometheus-Scrape-Konfiguration bezieht sich auf diese Datei.
- `docker_label` ist ein optionaler Abschnitt, mit dem Sie die Konfiguration für die Docker-Beschriftungs-basierte Service-Discovery angeben können. Wenn Sie diesen Abschnitt auslassen, wird die Docker-Bezeichnungs-basierte Erkennung nicht verwendet. Dieser Abschnitt kann die folgenden Felder enthalten:
  - `sd_port_label` ist der Docker-Bezeichnungsname des Containers, der den Container-Port für Prometheus Metriken angibt. Der Standardwert ist `ECS_PROMETHEUS_EXPORTER_PORT`. Wenn der Container dieses Docker-Label nicht hat, überspringt der CloudWatch Agent es.
  - `sd_metrics_path_label` ist der Docker-Bezeichnungsname des Containers, der den Pfad für Prometheus Metriken angibt. Der Standardwert ist `ECS_PROMETHEUS_METRICS_PATH`. Wenn der Container nicht über diese Docker-Bezeichnung verfügt, nimmt der Agent den Standardpfad `/metrics` an.
  - `sd_job_name_label` ist der Docker-Bezeichnungsname des Containers, der den Container-Scraping-Auftrag-Namen für Prometheus angibt. Der Standardwert ist `job`. Wenn der Container dieses Docker-Label nicht hat, verwendet der CloudWatch Agent den Jobnamen in der Prometheus-Scrape-Konfiguration.
- `task_definition_list` ist ein optionaler Abschnitt, den Sie verwenden können, um die Konfiguration der aufgabendefinitionsbasierten Serviceerkennung anzugeben. Wenn Sie diesen Abschnitt auslassen, wird die aufgabendefinitionsbasierte Erkennung nicht verwendet. Dieser Abschnitt kann die folgenden Felder enthalten:
  - `sd_task_definition_arn_pattern` ist das Muster, das verwendet wird, um die zu erkennenden Amazon-ECS-Aufgabendefinitionen anzugeben. Dies ist ein regulärer Ausdruck.
  - `sd_metrics_ports` listet den containerPort für die Prometheus-Metriken auf. Trennen Sie die ContainerPorts durch Semikolons.
  - `sd_container_name_pattern` gibt die Namen des Amazon-ECS-Aufgabencontainers an. Dies ist ein regulärer Ausdruck.
  - `sd_metrics_path` gibt den Prometheus-Metrikpfad an. Wenn Sie dies weglassen, übernimmt der Agent den Standardpfad `/metrics`
  - `sd_job_name` gibt den Namen des Prometheus -Scrape-Auftrags an. Wenn Sie dieses Feld weglassen, verwendet der CloudWatch Agent den Jobnamen in der Prometheus-Scrape-Konfiguration.
- `service_name_list_for_tasks` ist ein optionaler Abschnitt, den Sie verwenden können, um die Konfiguration der auf Servicenamen basierenden Erkennung anzugeben. Wenn Sie diesen

Abschnitt auslassen, wird die auf Servicenamen basierende Erkennung nicht verwendet. Dieser Abschnitt kann die folgenden Felder enthalten:

- `sd_service_name_pattern` ist das Muster, das verwendet werden soll, um den Amazon-ECS-Service anzugeben, in dem Aufgaben erkannt werden sollen. Dies ist ein regulärer Ausdruck.
- `sd_metrics_ports` listet den `containerPort` für die Prometheus-Metriken auf. Trennen Sie mehrere `containerPorts` durch Semikolons.
- `sd_container_name_pattern` gibt die Namen des Amazon-ECS-Aufgabencontainers an. Dies ist ein regulärer Ausdruck.
- `sd_metrics_path` gibt den Prometheus-Metrikpfad an. Wenn Sie dies weglassen, übernimmt der Agent den Standardpfad `/metrics`
- `sd_job_name` gibt den Namen des Prometheus -Scrape-Auftrags an. Wenn Sie dieses Feld weglassen, verwendet der CloudWatch Agent den Jobnamen in der Prometheus-Scrape-Konfiguration.

- metric_declaration – sind Abschnitte, die das Array von Protokollen mit eingebettetem Metrikformat angeben, das generiert werden soll. Für jede Prometheus-Quelle, aus der der CloudWatch Agent standardmäßig importiert, gibt es `metric_declaration` Abschnitte. Diese Abschnitte enthalten jeweils die folgenden Felder:
  - `label_matcher` ist ein regulärer Ausdruck, der den Wert der in `source_labels` aufgelisteten Beschriftungen überprüft. Die übereinstimmenden Metriken werden für die Aufnahme in das eingebettete Metrikformat aktiviert, an das gesendet wird. CloudWatch

    Wenn in `source_labels` mehrere Bezeichnungen angegeben sind, empfehlen wir, keine ^- oder $-Zeichen im regulären Ausdruck für `label_matcher` zu verwenden.
  - `source_labels` gibt den Wert der Beschriftungen an, die von der `label_matcher`-Zeile überprüft werden.
  - `label_separator` gibt das Trennzeichen an, das in der Zeile `label_matcher`verwendet werden soll, wenn mehrere `source_labels` angegeben werden. Der Standardwert ist `;`. Sie können diesen Standardwert in der Zeile `label_matcher` im folgenden Beispiel sehen.
  - `metric_selectors`ist ein regulärer Ausdruck, der die Metriken angibt, die gesammelt und an sie gesendet werden sollen CloudWatch.
  - `dimensions`ist die Liste der Bezeichnungen, die als CloudWatch Dimensionen für jede ausgewählte Metrik verwendet werden sollen.

Sehen Sie sich das folgende `metric_declaration`-Beispiel an.

```
"metric_declaration": [
  {
     "source_labels":[ "Service", "Namespace"],
     "label_matcher":"(.*node-exporter.*|.*kube-dns.*);kube-system$",
     "dimensions":[
        ["Service", "Namespace"]
     ],
     "metric_selectors":[
        "^coredns_dns_request_type_count_total$"
     ]
  }
]
```

In diesem Beispiel wird ein eingebetteter Metrikformatabschnitt konfiguriert, der als Protokollereignis gesendet wird, wenn die folgenden Bedingungen erfüllt sind:

- Der Wert von `Service` enthält entweder `node-exporter` oder `kube-dns`.
- Der Wert von `Namespace` ist `kube-system`.
- Die Prometheus-Metrik `coredns_dns_request_type_count_total` enthält sowohl `Service`- als auch `Namespace`-Beschriftungen.

Das Protokollereignis, das gesendet wird, enthält den folgenden hervorgehobenen Abschnitt:

```
{
   "CloudWatchMetrics":[
      {
         "Metrics":[
            {
               "Name":"coredns_dns_request_type_count_total"
            }
         ],
         "Dimensions":[
            [
               "Namespace",
               "Service"
            ]
         ],
         "Namespace":"ContainerInsights/Prometheus"
      }
```

```
    ],
    "Namespace":"kube-system",
    "Service":"kube-dns",
    "coredns_dns_request_type_count_total":2562,
    "eks_amazonaws_com_component":"kube-dns",
    "instance":"192.168.61.254:9153",
    "job":"kubernetes-service-endpoints",
    ...
}
```

Ausführliche Anleitung zu Autodiscovery auf Amazon-ECS-Clustern

Prometheus bietet Dutzende dynamischer Service-Discovery-Mechanismen, wie in <scrape_config> beschrieben. Es gibt jedoch keine integrierte Service-Erkennung für Amazon ECS. Der CloudWatch Agent fügt diesen Mechanismus hinzu.

Wenn die Amazon ECS Prometheus Service Discovery aktiviert ist, führt der CloudWatch Agent regelmäßig die folgenden API-Aufrufe an Amazon ECS- und Amazon EC2 EC2-Frontends durch, um die Metadaten der laufenden ECS-Aufgaben im ECS-Zielcluster abzurufen.

```
EC2:DescribeInstances
ECS:ListTasks
ECS:ListServices
ECS:DescribeContainerInstances
ECS:DescribeServices
ECS:DescribeTasks
ECS:DescribeTaskDefinition
```

Die Metadaten werden vom CloudWatch Agenten verwendet, um die Prometheus-Ziele innerhalb des ECS-Clusters zu scannen. Der CloudWatch Agent unterstützt drei Diensterkennungsmodi:

- Container-Docker-Label-basierte Service-Erkennung
- ECS-Aufgabendefinition, ARN basierte Service-Discovery mit regulären Ausdrücken
- ECS-Servicename, reguläre Ausdrucks-basierte Service-Erkennung

Alle Modi können zusammen verwendet werden. CloudWatch Der Agent dedupliziert die erkannten Ziele auf der Grundlage von:. `{private_ip}:{port}/{metrics_path}`

Alle erkannten Ziele werden in eine Ergebnisdatei geschrieben, die durch das `sd_result_file` Konfigurationsfeld im CloudWatch Agentencontainer angegeben wird. Das Folgende ist eine Beispielergebnisdatei:

```
- targets:
  - 10.6.1.95:32785
  labels:
    __metrics_path__: /metrics
    ECS_PROMETHEUS_EXPORTER_PORT: "9406"
    ECS_PROMETHEUS_JOB_NAME: demo-jar-ec2-bridge-dynamic
    ECS_PROMETHEUS_METRICS_PATH: /metrics
    InstanceType: t3.medium
    LaunchType: EC2
    SubnetId: subnet-123456789012
    TaskDefinitionFamily: demo-jar-ec2-bridge-dynamic-port
    TaskGroup: family:demo-jar-ec2-bridge-dynamic-port
    TaskRevision: "7"
    VpcId: vpc-01234567890
    container_name: demo-jar-ec2-bridge-dynamic-port
    job: demo-jar-ec2-bridge-dynamic
- targets:
  - 10.6.3.193:9404
  labels:
    __metrics_path__: /metrics
    ECS_PROMETHEUS_EXPORTER_PORT_SUBSET_B: "9404"
    ECS_PROMETHEUS_JOB_NAME: demo-tomcat-ec2-bridge-mapped-port
    ECS_PROMETHEUS_METRICS_PATH: /metrics
    InstanceType: t3.medium
    LaunchType: EC2
    SubnetId: subnet-123456789012
    TaskDefinitionFamily: demo-tomcat-ec2-bridge-mapped-port
    TaskGroup: family:demo-jar-tomcat-bridge-mapped-port
    TaskRevision: "12"
    VpcId: vpc-01234567890
    container_name: demo-tomcat-ec2-bridge-mapped-port
    job: demo-tomcat-ec2-bridge-mapped-port
```

Sie können diese Ergebnisdatei direkt in die dateibasierte Service-Erkennung von Prometheus integrieren. Weitere Informationen zur dateibasierten Serviceerkennung von Prometheus finden Sie unter <file_sd_config>

Angenommen, die Ergebnisdatei wird in `/tmp/cwagent_ecs_auto_sd.yaml` geschrieben. Die folgende Prometheus-Scrape-Konfiguration verbraucht sie.

```
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: cwagent-ecs-file-sd-config
    sample_limit: 10000
    file_sd_configs:
      - files: [ "/tmp/cwagent_ecs_auto_sd.yaml" ]
```

Der CloudWatch Agent fügt außerdem die folgenden zusätzlichen Bezeichnungen für die erkannten Ziele hinzu.

- `container_name`
- `TaskDefinitionFamily`
- `TaskRevision`
- `TaskGroup`
- `StartedBy`
- `LaunchType`
- `job`
- `__metrics_path__`
- Docker-Bezeichnungen

Wenn der Cluster den Starttyp EC2 hat, werden die folgenden drei Bezeichnungen hinzugefügt.

- `InstanceType`
- `VpcId`
- `SubnetId`

> **Note**
>
> Docker-Bezeichnungen, die nicht dem regulären Ausdruck `[a-zA-Z_][a-zA-Z0-9_]*` entsprechen, werden herausgefiltert. Dies stimmt mit den Prometheus-Konventionen

überein, die unter `label_name` in der Konfigurationsdatei in der Prometheus-Dokumentation aufgeführt sind.

Beispiele für die ECS-Serviceerkennung

Dieser Abschnitt enthält Beispiele, die die Ermittlung von ECS-Services veranschaulichen.

Beispiel 1

```
"ecs_service_discovery": {
  "sd_frequency": "1m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "docker_label": {
  }
}
```

In diesem Beispiel wird die Docker-Bezeichnungs-basierte Service-Discovery aktiviert. Der CloudWatch Agent fragt die Metadaten der ECS-Aufgaben einmal pro Minute ab und schreibt die erkannten Ziele in die `/tmp/cwagent_ecs_auto_sd.yaml` Datei im CloudWatch Agentencontainer.

Der Standardwert von `sd_port_label` im `docker_label`-Abschnitt ist `ECS_PROMETHEUS_EXPORTER_PORT`. Wenn ein laufender Container in den ECS-Aufgaben ein `ECS_PROMETHEUS_EXPORTER_PORT` Docker-Label hat, verwendet der CloudWatch Agent seinen Wert, `container port` um alle exponierten Ports des Containers zu scannen. Bei Übereinstimmung werden der zugeordnete Host-Port plus die private IP des Containers verwendet, um das Prometheus-Exporterziel im folgenden Format zu erstellen: `private_ip:host_port`.

Der Standardwert von `sd_metrics_path_label` im `docker_label`-Abschnitt ist `ECS_PROMETHEUS_METRICS_PATH`. Wenn der Container diese Docker-Bezeichnung hat, wird sein Wert als `__metrics_path__` verwendet. Wenn der Container diese Bezeichnung nicht hat, wird der Standardwert `/metrics` verwendet.

Der Standardwert von `sd_job_name_label` im `docker_label`-Abschnitt ist `job`. Wenn der Container über diese Docker-Bezeichnung verfügt, wird sein Wert als eine der Beschriftungen für das Ziel angehängt, um den in der Prometheus-Konfiguration angegebenen Standardauftragsnamen zu ersetzen. Der Wert dieses Docker-Labels wird als Log-Stream-Name in der CloudWatch Log-Protokollgruppe verwendet.

Beispiel 2

```
"ecs_service_discovery": {
  "sd_frequency": "15s",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "docker_label": {
    "sd_port_label": "ECS_PROMETHEUS_EXPORTER_PORT_SUBSET_A",
    "sd_job_name_label": "ECS_PROMETHEUS_JOB_NAME"
  }
}
```

In diesem Beispiel wird die Docker-Bezeichnungs-basierte Service-Discovery aktiviert. Der CloudWatch Agent fragt alle 15 Sekunden die Metadaten der ECS-Aufgaben ab und schreibt die erkannten Ziele in die `/tmp/cwagent_ecs_auto_sd.yaml` Datei im CloudWatch Agentencontainer. Die Container mit einer Docker-Bezeichnung von `ECS_PROMETHEUS_EXPORTER_PORT_SUBSET_A` werden gescannt. Der Wert der Docker-Bezeichnung `ECS_PROMETHEUS_JOB_NAME` wird als Auftragsname verwendet.

Beispiel 3

```
"ecs_service_discovery": {
  "sd_frequency": "5m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "task_definition_list": [
    {
      "sd_job_name": "java-prometheus",
      "sd_metrics_path": "/metrics",
      "sd_metrics_ports": "9404; 9406",
      "sd_task_definition_arn_pattern": ".*:task-definition/.*javajmx.*:[0-9]+"
    },
    {
      "sd_job_name": "envoy-prometheus",
      "sd_metrics_path": "/stats/prometheus",
      "sd_container_name_pattern": "^envoy$",
      "sd_metrics_ports": "9901",
      "sd_task_definition_arn_pattern": ".*:task-definition/.*appmesh.*:23"
    }
  ]
}
```

In diesem Beispiel wird die auf regulären Ausdrücken basierende Serviceerkennung des ECS-Aufgabendefinitions-ARN aktiviert. Der CloudWatch Agent fragt die Metadaten der ECS-Aufgaben

alle fünf Minuten ab und schreibt die erkannten Ziele in die `/tmp/cwagent_ecs_auto_sd.yaml` Datei im CloudWatch Agentencontainer.

Es werden zwei reguläre ARN Expresionsabschnitte für Aufgabendefinition definiert:

- Für den ersten Abschnitt werden die ECS-Aufgaben mit `javajmx` in ihrem ECS-Aufgabendefinitions-ARN für den Container-Port-Scan gefiltert. Wenn die Container innerhalb dieser ECS-Aufgaben den Container-Port auf 9404 oder 9406 verfügbar machen, werden der zugeordnete Host-Port zusammen mit der privaten IP-Adresse des Containers zum Erstellen der Prometheus-Exportziele verwendet. Der Wert von `sd_metrics_path` setzt `__metrics_path__` auf `/metrics`. Der CloudWatch Agent scrapt also die Prometheus-Metriken aus`private_ip:host_port/metrics`, die gescrapten Metriken werden an den Log-Stream unter CloudWatch Logs in der `java-prometheus` Log-Gruppe gesendet. `/aws/ecs/containerinsights/cluster_name/prometheus`
- Für den zweiten Abschnitt werden die ECS-Aufgaben mit `appmesh` in ihrem ECS-Aufgabendefinitions-ARN und mit `:23` von `version` für den Container-Port-Scan gefiltert. Für Container mit dem Namen `envoy` legen Sie den Container-Port auf `9901` offen. Der zugeordnete Host-Port wird zusammen mit der privaten IP des Containers verwendet, um die Prometheus-Exporterziele zu erstellen. Der Wert innerhalb dieser ECS-Aufgaben stellt den Container-Port auf 9404 oder 9406 zur Verfügung, der zugeordnete Host-Port zusammen mit der privaten IP des Containers werden verwendet, um die Prometheus-Exportziele zu erstellen. Der Wert von `sd_metrics_path` setzt `__metrics_path__` auf `/stats/prometheus`. Der CloudWatch Agent scrapt also die Prometheus-Metriken aus `private_ip:host_port/stats/prometheus` und sendet die gescrapten Metriken an den Log-Stream unter `envoy-prometheus` CloudWatch Logs in der Log-Gruppe. `/aws/ecs/containerinsights/cluster_name/prometheus`

Beispiel 4

```
"ecs_service_discovery": {
  "sd_frequency": "5m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "service_name_list_for_tasks": [
    {
      "sd_job_name": "nginx-prometheus",
      "sd_metrics_path": "/metrics",
      "sd_metrics_ports": "9113",
      "sd_service_name_pattern": "^nginx-.*"
    },
    {
```

```
      "sd_job_name": "haproxy-prometheus",
      "sd_metrics_path": "/stats/metrics",
      "sd_container_name_pattern": "^haproxy$",
      "sd_metrics_ports": "8404",
      "sd_service_name_pattern": ".*haproxy-service.*"
    }
  ]
}
```

In diesem Beispiel wird die auf regulären Ausdrücken basierende Serviceerkennung für ECS-Servicenamen aktiviert. Der CloudWatch Agent fragt alle fünf Minuten die Metadaten der ECS-Services ab und schreibt die erkannten Ziele in die `/tmp/cwagent_ecs_auto_sd.yaml` Datei im Agentencontainer. CloudWatch

Es werden zwei reguläre Expresionsabschnitte des Servicenamens definiert:

- Im ersten Abschnitt werden die ECS-Aufgaben, die ECS-Services zugeordnet sind, deren Namen mit dem regulären Ausdruck `^nginx-.*` übereinstimmen, für den Container-Port-Scan gefiltert Wenn die Container innerhalb dieser ECS-Aufgaben den Container-Port auf 9113 verfügbar machen, werden der zugeordnete Host-Port zusammen mit der privaten IP-Adresse des Containers zum Erstellen der Prometheus-Exportziele verwendet. Der Wert von `sd_metrics_path` setzt `__metrics_path__` auf `/metrics`. Der CloudWatch Agent scrapt also die Prometheus-Metriken aus`private_ip:host_port/metrics`, und die gescrapten Metriken werden an den `nginx-prometheus` Protokollstream unter CloudWatch Logs in der Protokollgruppe gesendet. `/aws/ecs/containerinsights/cluster_name/prometheus`
- Im zweiten Abschnitt werden die ECS-Aufgaben, die ECS-Services zugeordnet sind, deren Namen mit dem regulären Ausdruck `.*haproxy-service.*` übereinstimmen, für den Container-Port-Scan gefiltert Für Container mit dem Namen `haproxy` legen Sie den Container-Port auf 8404 offen. Der zugeordnete Host-Port wird zusammen mit der privaten IP des Containers verwendet, um die Prometheus-Exporterziele zu erstellen. Der Wert von `sd_metrics_path` setzt `__metrics_path__` auf `/stats/metrics`. Der CloudWatch Agent scrapt also die Prometheus-Metriken aus`private_ip:host_port/stats/metrics`, und die gescrapten Metriken werden an den `haproxy-prometheus` Protokollstream unter CloudWatch Logs in der Protokollgruppe gesendet. `/aws/ecs/containerinsights/cluster_name/prometheus`

Beispiel 5

```
"ecs_service_discovery": {
```

```
  "sd_frequency": "1m30s",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "docker_label": {
    "sd_port_label": "MY_PROMETHEUS_EXPORTER_PORT_LABEL",
    "sd_metrics_path_label": "MY_PROMETHEUS_METRICS_PATH_LABEL",
    "sd_job_name_label": "MY_PROMETHEUS_METRICS_NAME_LABEL"
  }
  "task_definition_list": [
    {
      "sd_metrics_ports": "9150",
      "sd_task_definition_arn_pattern": "*memcached.*"
    }
  ]
}
```

In diesem Beispiel werden beide ECS-Service-Ermittlungsmodi aktiviert. Der CloudWatch Agent fragt alle 90 Sekunden die Metadaten der ECS-Aufgaben ab und schreibt die erkannten Ziele in die `/tmp/cwagent_ecs_auto_sd.yaml` Datei im Agentencontainer. CloudWatch

Für die Docker-basierte Service-Discovery-Konfiguration:

- Die ECS-Aufgaben mit Docker-Bezeichnugn `MY_PROMETHEUS_EXPORTER_PORT_LABEL` werden für den Prometheus-Port-Scan gefiltert. Der Ziel-Prometheus-Container-Port wird durch den Wert der Bezeichnung `MY_PROMETHEUS_EXPORTER_PORT_LABEL` angegeben.
- Der Wert der Docker-Bezeichnung `MY_PROMETHEUS_EXPORTER_PORT_LABEL` wird für `__metrics_path__` verwendet. Wenn der Container diese Docker-Bezeichnung nicht hat, wird der Standardwert `/metrics` verwendet.
- Der Wert der Docker-Bezeichnung `MY_PROMETHEUS_EXPORTER_PORT_LABEL` wird als Auftragsbezeichnung verwendet. Wenn der Container nicht über diese Docker-Bezeichnung verfügt, wird der in der Prometheus-Konfiguration definierte Auftragsname verwendet.

Für die ECS-Aufgabendefinition ARN-basierte Serviceerkennungskonfiguration:

- Die ECS-Aufgaben mit `memcached` im ARN der ECS-Aufgabendefinition werden für den Container-Port-Scan gefiltert. Der Ziel-Container-Port von Prometheus ist 9150, wie durch `sd_metrics_ports` definiert. Der Standard-Metrikpfad `/metrics`wird verwendet. Der Auftragsname, der in der Prometheus-Konfiguration definiert ist, wird verwendet.

(Optional) Einrichten von containerisierten Beispiel-Amazon-ECS-Workloads für Prometheus-Metrik-Tests

Um die Unterstützung von Prometheus-Metriken in CloudWatch Container Insights zu testen, können Sie einen oder mehrere der folgenden containerisierten Workloads einrichten. Der CloudWatch Agent mit Prometheus-Unterstützung sammelt automatisch Metriken von jeder dieser Workloads. Informationen zum Anzeigen der Metriken, die standardmäßig erfasst werden, finden Sie unter Vom Agenten gesammelte Prometheus-Metriken CloudWatch .

Themen

- Beispiel für App-Mesh-Workload für Amazon-ECS-Cluster
- Beispiel für Java/JMX-Workload für Amazon-ECS-Cluster
- Beispiel für NGINX-Workload für Amazon ECS-Cluster
- Beispiel für NGINX-Plus-Workload für Amazon-ECS-Cluster
- Tutorial zum Hinzufügen eines neuen Prometheus-Scrape-Ziels: Memcached auf Amazon ECS
- Tutorial zum Scraping von Redis-Prometheus-Metriken auf Amazon ECS Fargate

Beispiel für App-Mesh-Workload für Amazon-ECS-Cluster

Um Metriken aus einem Prometheus-Beispiel-Workload für Amazon ECS zu erfassen, müssen Sie Container Insights im Cluster ausführen. Informationen zur Installation von Container Insights finden Sie unter Einrichten von Container Insights für Amazon ECS.

Befolgen Sie zunächst dieses Walkthrough, um die Beispielfarb-App auf Ihrem Amazon-ECS-Cluster bereitzustellen. Nachdem Sie fertig sind, werden App Mesh Prometheus-Metriken auf Port 9901 verfügbar gemacht.

Gehen Sie anschließend wie folgt vor, um den CloudWatch Agenten mit Prometheus-Überwachung auf demselben Amazon ECS-Cluster zu installieren, auf dem Sie die Farb-App installiert haben. Mit den Schritten in diesem Abschnitt wird der CloudWatch Agent im Bridge-Netzwerkmodus installiert.

Die Umgebungsvariablen `ENVIRONMENT_NAME`, `AWS_PROFILE` und `AWS_DEFAULT_REGION`, die Sie im Walkthrough festgelegt haben, werden auch in den folgenden Schritten verwendet.

Um den CloudWatch Agenten mit Prometheus-Überwachung zu Testzwecken zu installieren

1. Laden Sie die AWS CloudFormation Vorlage herunter, indem Sie den folgenden Befehl eingeben.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-
bridge-host.yaml
```

2. Legen Sie den Netzwerkmodus fest, indem Sie die folgenden Befehle eingeben.

```
export ECS_CLUSTER_NAME=${ENVIRONMENT_NAME}
export ECS_NETWORK_MODE=bridge
```

3. Erstellen Sie den AWS CloudFormation Stack, indem Sie die folgenden Befehle eingeben.

```
aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=True \
                 ParameterKey=ECSNetworkMode,ParameterValue=${ECS_NETWORK_MODE} \
                 ParameterKey=TaskRoleName,ParameterValue=CWAgent-Prometheus-
TaskRole-${ECS_CLUSTER_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=CWAgent-Prometheus-
ExecutionRole-${ECS_CLUSTER_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

4. (Optional) Wenn der AWS CloudFormation Stapel erstellt wird, wird eine `CREATE_COMPLETE` Meldung angezeigt. Wenn Sie den Status überprüfen möchten, bevor Sie diese Meldung sehen, geben Sie den folgenden Befehl ein.

```
aws cloudformation describe-stacks \
--stack-name CWAgent-Prometheus-ECS-${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
--query 'Stacks[0].StackStatus' \
--region ${AWS_DEFAULT_REGION} \
--profile ${AWS_PROFILE}
```

Fehlersuche

Die Schritte im Walkthrough verwenden jq, um das Ausgabeergebnis des AWS CLI zu analysieren. Weitere Informationen zum Installieren von jq finden Sie unter jq. Verwenden Sie den folgenden

Befehl, um das Standardausgabeformat Ihres AWS CLI auf JSON festzulegen, damit jq es richtig analysieren kann.

```
$ aws configure
```

Wenn die Antwort auf `Default output format` kommt, geben Sie den Wert **`json`** ein.

Deinstallieren Sie den CloudWatch Agenten mit Prometheus-Überwachung

Wenn Sie mit dem Testen fertig sind, geben Sie den folgenden Befehl ein, um den CloudWatch Agenten zu deinstallieren, indem Sie den AWS CloudFormation Stack löschen.

```
aws cloudformation delete-stack \
--stack-name CWAgent-Prometheus-ECS-${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
--region ${AWS_DEFAULT_REGION} \
--profile ${AWS_PROFILE}
```

Beispiel für Java/JMX-Workload für Amazon-ECS-Cluster

JMX Exporter ist ein offizieller Prometheus-Exporter, der JMX mBeans als Prometheus-Metriken erfassen und verfügbar machen kann. Weitere Informationen finden Sie unter prometheus/jmx_exporter.

Der CloudWatch Agent mit Prometheus-Unterstützung scannt die Java/JMX Prometheus-Metriken auf der Grundlage der Service Discovery-Konfiguration im Amazon ECS-Cluster. Sie können den JMX Exporter so konfigurieren, dass die Metriken auf einem anderen Port oder metrics_path verfügbar gemacht werden. Wenn Sie den Port oder Pfad ändern, aktualisieren Sie den Standardabschnitt in der Agentenkonfiguration. `ecs_service_discovery` CloudWatch

Um Metriken aus einem Prometheus-Beispiel-Workload für Amazon ECS zu erfassen, müssen Sie Container Insights im Cluster ausführen. Informationen zur Installation von Container Insights finden Sie unter Einrichten von Container Insights für Amazon ECS.

So installieren Sie die Java/JMX-Beispiel-Workload für Amazon-ECS-Cluster

1. Führen Sie die Schritte in diesen Abschnitten aus, um Ihre Docker-Images zu erstellen.

   - Beispiel: Docker-Image der Java-Jar-Anwendung mit Prometheus-Metriken
   - Beispiel: Apache-Tomcat-Docker-Image mit Prometheus-Metriken

2. Geben Sie die folgenden zwei Docker-Bezeichnungen in der Amazon-ECS-Aufgabendefinitionsdatei an. Anschließend können Sie die Aufgabendefinition als Amazon-ECS-Service oder Amazon-ECS-Aufgabe im Cluster ausführen.

   - Legen Sie `ECS_PROMETHEUS_EXPORTER_PORT` so fest, dass es auf den ContainerPort zeigt, in dem die Prometheus-Metriken verfügbar gemacht werden.
   - Setzen Sie `Java_EMF_Metrics` auf `true`. Der CloudWatch Agent verwendet dieses Flag, um das eingebettete metrische Format im Protokollereignis zu generieren.

   Im Folgenden wird ein Beispiel gezeigt:

```
{
  "family": "workload-java-ec2-bridge",
  "taskRoleArn": "{{task-role-arn}}",
  "executionRoleArn": "{{execution-role-arn}}",
  "networkMode": "bridge",
  "containerDefinitions": [
    {
      "name": "tomcat-prometheus-workload-java-ec2-bridge-dynamic-port",
      "image": "your_docker_image_tag_for_tomcat_with_prometheus_metrics",
      "portMappings": [
        {
          "hostPort": 0,
          "protocol": "tcp",
          "containerPort": 9404
        }
      ],
      "dockerLabels": {
        "ECS_PROMETHEUS_EXPORTER_PORT": "9404",
        "Java_EMF_Metrics": "true"
      }
    }
  ],
  "requiresCompatibilities": [
    "EC2"  ],
  "cpu": "256",
  "memory": "512"
  }
```

Die Standardeinstellung des CloudWatch Agenten in der AWS CloudFormation Vorlage ermöglicht sowohl die auf Docker-Labels basierende Diensterkennung als auch die ARN-basierte Diensterkennung mit Aufgabendefinitionen. Diese Standardeinstellungen finden Sie in Zeile 65 der YAML-Konfigurationsdatei für den CloudWatch Agenten. Die Container mit der `ECS_PROMETHEUS_EXPORTER_PORT`-Bezeichnung werden basierend auf dem angegebenen Container-Port für das Prometheus-Scraping automatisch erkannt.

Die Standardeinstellung des CloudWatch Agenten enthält auch die `metric_declaration` Einstellung für Java/JMX in Zeile 112 derselben Datei. Alle Docker-Labels der Zielcontainer werden als zusätzliche Labels zu den Prometheus-Metriken hinzugefügt und an Logs gesendet. CloudWatch Für die Java/JMX-Container mit Docker-Bezeichnung `Java_EMF_Metrics="true"` wird das eingebettete Metrikformat generiert.

Beispiel für NGINX-Workload für Amazon ECS-Cluster

Der NGINX Prometheus Exporter kann NGINX-Daten als Prometheus-Metriken scrapen und verfügbar machen. In diesem Beispiel wird der Exporter zusammen mit dem NGINX-Reverse-Proxy-Service für Amazon ECS verwendet.

Weitere Informationen zum NGINX Prometheus Exporter finden Sie auf Github. nginx-prometheus-exporter Weitere Informationen zum NGINX-Reverse-Proxy finden Sie auf Github. ecs-nginx-reverse-proxy

Der CloudWatch Agent mit Prometheus-Unterstützung scannt die NGINX Prometheus-Metriken auf der Grundlage der Service Discovery-Konfiguration im Amazon ECS-Cluster. Sie können den NGINX Prometheus Exporter so konfigurieren, dass die Metriken auf einem anderen Port oder metrics_path verfügbar gemacht werden. Wenn Sie den Port oder Pfad ändern, aktualisieren Sie den Abschnitt in der `ecs_service_discovery` Agenten-Konfigurationsdatei. CloudWatch

Installieren Sie den NGINX-Reverse-Proxy-Beispiel-Workload für Amazon ECS-Cluster

Zum Installieren des NGINX-Reverse-Proxy-Beispiel-Workloads führen Sie die folgenden Schritte aus.

Erstellen der Docker-Images

So erstellen Sie die Docker-Images für die NGINX-Reverse-Proxy-Beispiel-Workload

1. Laden Sie den folgenden Ordner aus dem NGINX-Reverse-Proxy-Repo herunter: https://github.com/awslabs/ /tree/master/reverse-proxy/. ecs-nginx-reverse-proxy

2. Suchen Sie das app-Verzeichnis und erstellen Sie ein Image aus diesem Verzeichnis:

```
docker build -t web-server-app ./path-to-app-directory
```

3. Erstellen Sie ein benutzerdefiniertes Image für NGINX. Erstellen Sie zunächst ein Verzeichnis mit den folgenden zwei Dateien:

   - Eine Beispieldatei für Dockerdatei:

```
FROM nginx
COPY nginx.conf /etc/nginx/nginx.conf
```

   - Eine `nginx.conf` Datei, geändert von ecs-nginx-reverse-proxyhttps://github.com/awslabs/ / tree/master/reverse-proxy/:

```
events {
  worker_connections 768;
}

http {
  # Nginx will handle gzip compression of responses from the app server
  gzip on;
  gzip_proxied any;
  gzip_types text/plain application/json;
  gzip_min_length 1000;

  server{
    listen 8080;
    location /stub_status {
        stub_status   on;
    }
  }

  server {
    listen 80;

    # Nginx will reject anything not matching /api
    location /api {
      # Reject requests with unsupported HTTP method
      if ($request_method !~ ^(GET|POST|HEAD|OPTIONS|PUT|DELETE)$) {
        return 405;
      }
```

```
      # Only requests matching the whitelist expectations will
      # get sent to the application server
      proxy_pass http://app:3000;
      proxy_http_version 1.1;
      proxy_set_header Upgrade $http_upgrade;
      proxy_set_header Connection 'upgrade';
      proxy_set_header Host $host;
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_cache_bypass $http_upgrade;
    }
  }
}
```

> **Note**
>
> `stub_status` muss auf demselben Port aktiviert sein, für den `nginx-prometheus-exporter` konfiguriert ist, um Metriken zu scrapen. In unserer Beispielaufgabendefinition ist `nginx-prometheus-exporter` so konfiguriert, dass Metriken von Port 8080 gescrapet werden.

4. Erstellen Sie ein Image aus Dateien in Ihrem neuen Verzeichnis:

```
docker build -t nginx-reverse-proxy ./path-to-your-directory
```

5. Laden Sie Ihre neuen Images zur späteren Verwendung in ein Bild-Repository hoch.

Erstellen Sie die Aufgabendefinition zum Ausführen von NGINX und der Webserver-App in Amazon ECS

Als Nächstes richten Sie die Aufgabendefinition ein.

Diese Aufgabendefinition ermöglicht das Sammeln und Exportieren von NGINX-Prometheus-Metriken. Der NGINX-Container verfolgt die Eingaben von der App und stellt diese Daten an Port 8080 bereit, wie in `nginx.conf` festgelegt. Der NGINX-Prometheus-Exporter-Container scannt diese Metriken und sendet sie zur Verwendung in Port 9113. CloudWatch

So richten Sie die Aufgabendefinition für die NGINX-Beispiel-Amazon ECS-Workload ein

1. Erstellen Sie eine Aufgabendefinitions-JSON-Datei mit dem folgenden Inhalt. *your-customized-nginx-iamge*Ersetzen Sie sie durch die Bild-URI für Ihr benutzerdefiniertes

NGINX-Image und ersetzen Sie -image durch die Bild-URI für Ihr *your-web-server-appWebserver-App-Image*.

```
{
  "containerDefinitions": [
    {
      "name": "nginx",
      "image": "your-customized-nginx-image",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "portMappings": [
        {
          "containerPort": 80,
          "protocol": "tcp"
        }
      ],
      "links": [
        "app"
      ]
    },
    {
      "name": "app",
      "image": "your-web-server-app-image",
      "memory": 256,
      "cpu": 256,
      "essential": true
    },
    {
      "name": "nginx-prometheus-exporter",
      "image": "docker.io/nginx/nginx-prometheus-exporter:0.8.0",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "command": [
        "-nginx.scrape-uri",
        "http://nginx:8080/stub_status"
    ],
    "links":[
      "nginx"
    ],
      "portMappings":[
        {
```

```
          "containerPort": 9113,
          "protocol": "tcp"
        }
      ]
    }
  ],
  "networkMode": "bridge",
  "placementConstraints": [],
  "family": "nginx-sample-stack"
}
```

2. Registrieren Sie die Aufgabendefinition, indem Sie den folgenden Befehl eingeben.

```
aws ecs register-task-definition --cli-input-json file://path-to-your-task-definition-json
```

3. Erstellen Sie einen Service zum Ausführen der Aufgabe, indem Sie den folgenden Befehl eingeben:

   Stellen Sie sicher, dass Sie den Servicenamen nicht ändern. Wir werden einen CloudWatch Agentendienst mit einer Konfiguration ausführen, die anhand der Namensmuster der Dienste, die sie gestartet haben, nach Aufgaben sucht. Damit der CloudWatch Agent beispielsweise die mit diesem Befehl gestartete Aufgabe finden kann, können Sie den Wert `sd_service_name_pattern` to be angeben`^nginx-service$`. Im nächsten Abschnitt finden Sie weitere Details.

```
aws ecs create-service \
 --cluster your-cluster-name \
 --service-name nginx-service \
 --task-definition nginx-sample-stack:1 \
 --desired-count 1
```

Konfigurieren Sie den CloudWatch Agenten für das Scrapen von NGINX Prometheus-Metriken

Der letzte Schritt besteht darin, den CloudWatch Agenten so zu konfigurieren, dass er die NGINX-Metriken scannt. In diesem Beispiel erkennt der CloudWatch Agent die Aufgabe anhand des Dienstnamenmusters und des Ports 9113, wo der Exporter die Prometheus-Metriken für NGINX verfügbar macht. Sobald die Aufgabe erkannt wurde und die Metriken verfügbar sind, beginnt der CloudWatch Agent, die gesammelten Metriken im Log-Stream zu posten. nginx-prometheus-exporter

Um den CloudWatch Agenten so zu konfigurieren, dass er die NGINX-Metriken scannt

1. Laden Sie die neueste Version der erforderlichen YAML-Datei herunter, indem Sie den folgenden Befehl eingeben.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-bridge-host.yaml
```

2. Öffnen Sie die Datei mit einem Texteditor und finden Sie die vollständige CloudWatch Agentenkonfiguration im Schlüssel im `value` Abschnitt. `resource:CWAgentConfigSSMParameter` Fügen Sie dann im `ecs_service_discovery`-Abschnitt den folgenden `service_name_list_for_tasks`-Abschnitt hinzu.

```
"service_name_list_for_tasks": [
  {
    "sd_job_name": "nginx-prometheus-exporter",
    "sd_metrics_path": "/metrics",
    "sd_metrics_ports": "9113",
    "sd_service_name_pattern": "^nginx-service$"
   }
],
```

3. Fügen Sie in derselben Datei den folgenden Abschnitt im Abschnitt `metric_declaration` hinzu, um NGINX-Metriken zuzulassen. Beachten Sie unbedingt das vorhandene Einrückungsmuster.

```
{
  "source_labels": ["job"],
  "label_matcher": ".*nginx.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName"]],
  "metric_selectors": [
    "^nginx_.*$"
  ]
},
```

4. Wenn Sie den CloudWatch Agenten noch nicht in diesem Cluster bereitgestellt haben, fahren Sie mit Schritt 8 fort.

Wenn Sie den CloudWatch Agenten bereits mithilfe von im Amazon ECS-Cluster bereitgestellt haben AWS CloudFormation, können Sie einen Änderungssatz erstellen, indem Sie die folgenden Befehle eingeben:

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
    --change-set-name nginx-scraping-support
```

5. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
6. Überprüfen Sie das neu erstellte Changeset. nginx-scraping-support Sie sollten sehen, dass eine Änderung auf die CW-SSMParameter-Ressource angewendet wurde. AgentConfig Führen Sie das Changeset aus und starten Sie die CloudWatch Agententask neu, indem Sie den folgenden Befehl eingeben:

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

7. Warten Sie etwa 10 Sekunden und geben Sie den folgenden Befehl ein.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
```

```
--region $AWS_REGION
```

8. Wenn Sie den CloudWatch Agenten mit der Erfassung von Prometheus-Metriken zum ersten Mal auf dem Cluster installieren, geben Sie die folgenden Befehle ein.

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

Anzeigen Ihrer NGINX-Metriken und Protokolle

Sie können jetzt die gesammelten NGINX-Metriken anzeigen.

So zeigen Sie die Metriken für Ihren NGINX-Beispiel-Workload an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie in der Region, in der Ihr Cluster ausgeführt wird, im linken Navigationsbereich Metriken aus. Suchen Sie den ContainerInsights/Prometheus-Namespace, um die Metriken zu sehen.
3. Um die CloudWatch Logs-Ereignisse zu sehen, wählen Sie im Navigationsbereich Protokollgruppen aus. *Die Ereignisse befinden sich in der Protokollgruppe **/aws/containerinsights/** your_cluster_name /prometheus im Protokollstream. nginx-prometheus-exporter*

Beispiel für NGINX-Plus-Workload für Amazon-ECS-Cluster

NGINX Plus ist die kommerzielle Version von NGINX. Sie müssen über eine Lizenz verfügen, um sie zu verwenden. Weitere Informationen finden Sie unter NGINX Plus.

Der NGINX Prometheus Exporter kann NGINX-Daten als Prometheus-Metriken scrapen und verfügbar machen. In diesem Beispiel wird der Exporter zusammen mit dem NGINX-Plus-Reverse-Proxy-Service für Amazon ECS verwendet.

Weitere Informationen zum NGINX Prometheus Exporter finden Sie auf Github. nginx-prometheus-exporter Weitere Informationen zum NGINX-Reverse-Proxy finden Sie auf Github. ecs-nginx-reverse-proxy

Der CloudWatch Agent mit Prometheus-Unterstützung scannt die NGINX Plus Prometheus-Metriken auf der Grundlage der Service Discovery-Konfiguration im Amazon ECS-Cluster. Sie können den NGINX Prometheus Exporter so konfigurieren, dass die Metriken auf einem anderen Port oder metrics_path verfügbar gemacht werden. Wenn Sie den Port oder Pfad ändern, aktualisieren Sie den Abschnitt in der `ecs_service_discovery` Agenten-Konfigurationsdatei. CloudWatch

Installieren Sie den NGINX-Plus-Reverse-Proxy-Beispiel-Workload für Amazon-ECS-Cluster

Zum Installieren des NGINX-Reverse-Proxy-Beispiel-Workloads führen Sie die folgenden Schritte aus.

Erstellen der Docker-Images

So erstellen Sie die Docker-Images für die NGINX-Plus-Reverse-Proxy-Beispiel-Workload

1. Laden Sie den folgenden Ordner aus dem NGINX-Reverse-Proxy-Repo herunter: https://github.com/awslabs/ /tree/master/reverse-proxy/. ecs-nginx-reverse-proxy
2. Suchen Sie das app-Verzeichnis und erstellen Sie ein Image aus diesem Verzeichnis:

   ```
   docker build -t web-server-app ./path-to-app-directory
   ```

3. Erstellen Sie ein benutzerdefiniertes Image für NGINX Plus. Bevor Sie das Image für NGINX Plus erstellen können, benötigen Sie den Schlüssel mit dem Namen `nginx-repo.key` und das SSL-Zertifikat `nginx-repo.crt` für Ihr lizenziertes NGINX Plus. Erstellen Sie ein Verzeichnis und speichern Sie darin Ihre `nginx-repo.key`- und `nginx-repo.crt`-Dateien.

   Erstellen Sie in dem Verzeichnis, das Sie gerade erstellt haben, die folgenden zwei Dateien:

- Ein Beispiel-Dockerfile mit dem folgenden Inhalt. Diese Docker-Datei wurde aus einer Beispieldatei übernommen, die unter https://docs.nginx.com/nginx/admin-guide/installing-nginx/installing-nginx-docker/#docker_plus_image bereitgestellt wird. Die wichtige Änderung, die wir vornehmen, ist, dass wir eine separate Datei namens `nginx.conf` laden, die im nächsten Schritt erstellt wird.

```
FROM debian:buster-slim

LABEL maintainer="NGINX Docker Maintainers <docker-maint@nginx.com>“

# Define NGINX versions for NGINX Plus and NGINX Plus modules
# Uncomment this block and the versioned nginxPackages block in the main RUN
# instruction to install a specific release
# ENV NGINX_VERSION 21
# ENV NJS_VERSION 0.3.9
# ENV PKG_RELEASE 1~buster

# Download certificate and key from the customer portal (https://cs.nginx.com
 (https://cs.nginx.com/))
# and copy to the build context
COPY nginx-repo.crt /etc/ssl/nginx/
COPY nginx-repo.key /etc/ssl/nginx/
# COPY nginx.conf /etc/ssl/nginx/nginx.conf

RUN set -x \
# Create nginx user/group first, to be consistent throughout Docker variants
&& addgroup --system --gid 101 nginx \
&& adduser --system --disabled-login --ingroup nginx --no-create-home --home /
nonexistent --gecos "nginx user" --shell /bin/false --uid 101 nginx \
&& apt-get update \
&& apt-get install --no-install-recommends --no-install-suggests -y ca-
certificates gnupg1 \
&& \
NGINX_GPGKEY=573BFD6B3D8FBC641079A6ABABF5BD827BD9BF62; \
found=''; \
for server in \
ha.pool.sks-keyservers.net (http://ha.pool.sks-keyservers.net/) \
hkp://keyserver.ubuntu.com:80 \
hkp://p80.pool.sks-keyservers.net:80 \
pgp.mit.edu (http://pgp.mit.edu/) \
; do \
echo "Fetching GPG key $NGINX_GPGKEY from $server"; \
```

```
apt-key adv --keyserver "$server" --keyserver-options timeout=10 --recv-keys
 "$NGINX_GPGKEY" && found=yes && break; \
done; \
test -z "$found" && echo >&2 "error: failed to fetch GPG key $NGINX_GPGKEY" &&
 exit 1; \
apt-get remove --purge --auto-remove -y gnupg1 && rm -rf /var/lib/apt/lists/* \
# Install the latest release of NGINX Plus and/or NGINX Plus modules
# Uncomment individual modules if necessary
# Use versioned packages over defaults to specify a release
&& nginxPackages=" \
nginx-plus \
# nginx-plus=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-xslt \
# nginx-plus-module-xslt=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-geoip \
# nginx-plus-module-geoip=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-image-filter \
# nginx-plus-module-image-filter=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-perl \
# nginx-plus-module-perl=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-njs \
# nginx-plus-module-njs=${NGINX_VERSION}+${NJS_VERSION}-${PKG_RELEASE} \
" \
&& echo "Acquire::https::plus-pkgs.nginx.com::Verify-Peer \"true\";" >> /etc/apt/
apt.conf.d/90nginx \
&& echo "Acquire::https::plus-pkgs.nginx.com::Verify-Host \"true\";" >> /etc/apt/
apt.conf.d/90nginx \
&& echo "Acquire::https::plus-pkgs.nginx.com::SslCert \"/etc/ssl/nginx/nginx-
repo.crt\";" >> /etc/apt/apt.conf.d/90nginx \
&& echo "Acquire::https::plus-pkgs.nginx.com::SslKey \"/etc/ssl/nginx/nginx-
repo.key\";" >> /etc/apt/apt.conf.d/90nginx \
&& printf "deb https://plus-pkgs.nginx.com/debian buster nginx-plus\n" > /etc/
apt/sources.list.d/nginx-plus.list \
&& apt-get update \
&& apt-get install --no-install-recommends --no-install-suggests -y \
$nginxPackages \
gettext-base \
curl \
&& apt-get remove --purge --auto-remove -y && rm -rf /var/lib/apt/lists/* /etc/
apt/sources.list.d/nginx-plus.list \
&& rm -rf /etc/apt/apt.conf.d/90nginx /etc/ssl/nginx

# Forward request logs to Docker log collector
RUN ln -sf /dev/stdout /var/log/nginx/access.log \
```

```
&& ln -sf /dev/stderr /var/log/nginx/error.log

COPY nginx.conf /etc/nginx/nginx.conf

EXPOSE 80

STOPSIGNAL SIGTERM

CMD ["nginx", "-g", "daemon off;"]
```

- Eine `nginx.conf` Datei, geändert von https://github.com/awslabs/ ecs-nginx-reverse-proxy / tree/master/reverse-proxy/nginx.

```
events {
  worker_connections 768;
}

http {
  # Nginx will handle gzip compression of responses from the app server
  gzip on;
  gzip_proxied any;
  gzip_types text/plain application/json;
  gzip_min_length 1000;

  upstream backend {
    zone name 10m;
    server app:3000    weight=2;
    server app2:3000    weight=1;
  }

  server{
    listen 8080;
    location /api {
      api write=on;
    }
  }

  match server_ok {
    status 100-599;
  }

  server {
    listen 80;
```

```
    status_zone zone;
    # Nginx will reject anything not matching /api
    location /api {
      # Reject requests with unsupported HTTP method
      if ($request_method !~ ^(GET|POST|HEAD|OPTIONS|PUT|DELETE)$) {
        return 405;
      }

      # Only requests matching the whitelist expectations will
      # get sent to the application server
      proxy_pass http://backend;
      health_check uri=/lorem-ipsum match=server_ok;
      proxy_http_version 1.1;
      proxy_set_header Upgrade $http_upgrade;
      proxy_set_header Connection 'upgrade';
      proxy_set_header Host $host;
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_cache_bypass $http_upgrade;
    }
  }
}
```

4. Erstellen Sie ein Image aus Dateien in Ihrem neuen Verzeichnis:

```
docker build -t nginx-plus-reverse-proxy ./path-to-your-directory
```

5. Laden Sie Ihre neuen Images zur späteren Verwendung in ein Bild-Repository hoch.

Erstellen Sie die Aufgabendefinition zum Ausführen von NGINX Plus und der Webserver-App in Amazon ECS

Als Nächstes richten Sie die Aufgabendefinition ein.

Diese Aufgabendefinition ermöglicht das Sammeln und Exportieren von NGINX-Plus-Prometheus-Metriken. Der NGINX-Container verfolgt die Eingaben von der App und stellt diese Daten an Port 8080 bereit, wie in `nginx.conf` festgelegt. Der NGINX-Prometheus-Exporter-Container scannt diese Metriken und sendet sie zur Verwendung in Port 9113. CloudWatch

So richten Sie die Aufgabendefinition für die NGINX-Beispiel-Amazon ECS-Workload ein

1. Erstellen Sie eine Aufgabendefinitions-JSON-Datei mit dem folgenden Inhalt. Ersetzen Sie *your-customized-nginx-plus-image* durch den Bild-URI für Ihr benutzerdefiniertes

NGINX Plus-Image und ersetzen Sie -image durch den Bild-URI für Ihr *your-web-server-appWebserver-App-Image*.

```
{
  "containerDefinitions": [
    {
      "name": "nginx",
      "image": "your-customized-nginx-plus-image",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "portMappings": [
        {
          "containerPort": 80,
          "protocol": "tcp"
        }
      ],
      "links": [
        "app",
        "app2"
      ]
    },
    {
      "name": "app",
      "image": "your-web-server-app-image",
      "memory": 256,
      "cpu": 128,
      "essential": true
    },
    {
      "name": "app2",
      "image": "your-web-server-app-image",
      "memory": 256,
      "cpu": 128,
      "essential": true
    },
    {
      "name": "nginx-prometheus-exporter",
      "image": "docker.io/nginx/nginx-prometheus-exporter:0.8.0",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "command": [
```

```
        "-nginx.plus",
        "-nginx.scrape-uri",
         "http://nginx:8080/api"
    ],
    "links":[
      "nginx"
    ],
      "portMappings":[
        {
          "containerPort": 9113,
          "protocol": "tcp"
        }
      ]
    }
  ],
  "networkMode": "bridge",
  "placementConstraints": [],
  "family": "nginx-plus-sample-stack"
}
```

2. Registrieren Sie die Aufgabendefinition:

```
aws ecs register-task-definition --cli-input-json file://path-to-your-task-definition-json
```

3. Erstellen Sie einen Service zum Ausführen der Aufgabe, indem Sie den folgenden Befehl eingeben:

```
aws ecs create-service \
 --cluster your-cluster-name \
 --service-name nginx-plus-service \
 --task-definition nginx-plus-sample-stack:1 \
 --desired-count 1
```

Stellen Sie sicher, dass Sie den Servicenamen nicht ändern. Wir werden einen CloudWatch Agentendienst mit einer Konfiguration ausführen, die anhand der Namensmuster der Dienste, die sie gestartet haben, nach Aufgaben sucht. Damit der CloudWatch Agent beispielsweise die mit diesem Befehl gestartete Aufgabe finden kann, können Sie den Wert `sd_service_name_pattern` to be angeben`^nginx-plus-service$`. Im nächsten Abschnitt finden Sie weitere Details.

Konfigurieren Sie den CloudWatch Agenten für das Scraping von NGINX Plus Prometheus-Metriken

Der letzte Schritt besteht darin, den CloudWatch Agenten so zu konfigurieren, dass er die NGINX-Metriken scannt. In diesem Beispiel erkennt der CloudWatch Agent die Aufgabe anhand des Dienstnamenmusters und des Ports 9113, wo der Exporter die Prometheus-Metriken für NGINX verfügbar macht. Sobald die Aufgabe erkannt wurde und die Metriken verfügbar sind, beginnt der CloudWatch Agent, die gesammelten Metriken im Log-Stream zu posten. nginx-prometheus-exporter

Um den CloudWatch Agenten so zu konfigurieren, dass er die NGINX-Metriken scannt

1. Laden Sie die neueste Version der erforderlichen YAML-Datei herunter, indem Sie den folgenden Befehl eingeben.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-bridge-host.yaml
```

2. Öffnen Sie die Datei mit einem Texteditor und finden Sie die vollständige CloudWatch Agentenkonfiguration im Schlüssel im `value` Abschnitt. `resource:CWAgentConfigSSMParameter` Fügen Sie dann im `ecs_service_discovery`-Abschnitt den folgenden `service_name_list_for_tasks`-Abschnitt hinzu.

```
"service_name_list_for_tasks": [
  {
    "sd_job_name": "nginx-plus-prometheus-exporter",
    "sd_metrics_path": "/metrics",
    "sd_metrics_ports": "9113",
    "sd_service_name_pattern": "^nginx-plus.*"
   }
],
```

3. Fügen Sie in derselben Datei den folgenden Abschnitt im Abschnitt `metric_declaration` hinzu, um NGINX-Plus-Metriken zuzulassen. Beachten Sie unbedingt das vorhandene Einrückungsmuster.

```
{
  "source_labels": ["job"],
  "label_matcher": "^nginx-plus.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName"]],
  "metric_selectors": [
```

```
    "^nginxplus_connections_accepted$",
    "^nginxplus_connections_active$",
    "^nginxplus_connections_dropped$",
    "^nginxplus_connections_idle$",
    "^nginxplus_http_requests_total$",
    "^nginxplus_ssl_handshakes$",
    "^nginxplus_ssl_handshakes_failed$",
    "^nginxplus_up$",
    "^nginxplus_upstream_server_health_checks_fails$"
  ]
},
{
  "source_labels": ["job"],
  "label_matcher": "^nginx-plus.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName",
 "upstream"]],
  "metric_selectors": [
    "^nginxplus_upstream_server_response_time$"
  ]
},
{
  "source_labels": ["job"],
  "label_matcher": "^nginx-plus.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName", "code"]],
  "metric_selectors": [
    "^nginxplus_upstream_server_responses$",
    "^nginxplus_server_zone_responses$"
  ]
},
```

4. Wenn Sie den CloudWatch Agenten noch nicht in diesem Cluster bereitgestellt haben, fahren Sie mit Schritt 8 fort.

   Wenn Sie den CloudWatch Agenten bereits mithilfe von im Amazon ECS-Cluster bereitgestellt haben AWS CloudFormation, können Sie einen Änderungssatz erstellen, indem Sie die folgenden Befehle eingeben:

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name
```

```
aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
    --change-set-name nginx-plus-scraping-support
```

5. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
6. Überprüfen Sie das neu erstellte Changeset. nginx-plus-scraping-support Sie sollten sehen, dass eine Änderung auf die CW-SSMParameter-Ressource angewendet wurde. AgentConfig Führen Sie das Changeset aus und starten Sie die CloudWatch Agententask neu, indem Sie den folgenden Befehl eingeben:

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

7. Warten Sie etwa 10 Sekunden und geben Sie den folgenden Befehl ein.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

8. Wenn Sie den CloudWatch Agenten mit der Erfassung von Prometheus-Metriken zum ersten Mal auf dem Cluster installieren, geben Sie die folgenden Befehle ein.

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name
```

```
aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

Anzeigen Ihrer NGINX-Plus-Metriken und -Protokolle

Sie können jetzt die gesammelten NGINX-Plus-Metriken anzeigen.

So zeigen Sie die Metriken für Ihren NGINX-Beispiel-Workload an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie in der Region, in der Ihr Cluster ausgeführt wird, im linken Navigationsbereich Metriken aus. Suchen Sie den ContainerInsights/Prometheus-Namespace, um die Metriken zu sehen.
3. Um die CloudWatch Logs-Ereignisse zu sehen, wählen Sie im Navigationsbereich Protokollgruppen aus. *Die Ereignisse befinden sich in der Protokollgruppe* ***/aws/containerinsights/*** *your_cluster_name /prometheus im Protokollstream. nginx-plus-prometheus-exporter*

Tutorial zum Hinzufügen eines neuen Prometheus-Scrape-Ziels: Memcached auf Amazon ECS

Dieses Tutorial bietet eine praktische Einführung in die Prometheus-Metriken einer Memcached-Beispielanwendung auf einem Amazon-ECS-Cluster mit dem EC2-Starttyp. Das Memcached Prometheus-Exporter-Ziel wird vom CloudWatch Agenten automatisch durch die auf ECS-Aufgabendefinitionen basierende Serviceerkennung erkannt.

Memcached ist ein universelles verteiltes Speicherzwischenspeicher-Caching-System. Es wird häufig verwendet, um dynamische datenbankgesteuerte Websites zu beschleunigen, indem Daten und Objekte im RAM zwischengespeichert werden, um die Häufigkeit des Lesens einer externen

Datenquelle (z. B. einer Datenbank oder API) zu reduzieren. Weitere Informationen finden Sie unter Was ist Memcached?.

Der memchached_exporter (Apache License 2.0) ist einer der offiziellen Prometheus-Exporteure. Standardmäßig dient der memcache_exporter auf Port 0.0.0.0:9150 bei `/metrics`.

Die Docker-Images in den folgenden zwei Docker Hub-Repositories werden in diesem Tutorial verwendet:

- Memcached
- prom/memcached-exporter

Voraussetzung

Um Metriken aus einem Prometheus-Beispiel-Workload für Amazon ECS zu erfassen, müssen Sie Container Insights im Cluster ausführen. Informationen zur Installation von Container Insights finden Sie unter Einrichten von Container Insights für Amazon ECS.

Themen

- Legen Sie die Umgebungsvariablen des EC2-Clusters von Amazon ECS fest
- Installieren der Memcached-Beispiel-Workload
- Konfigurieren Sie den CloudWatch Agenten für das Scraping von Memcached Prometheus-Metriken
- Anzeigen Ihrer Memcached-Metriken

Legen Sie die Umgebungsvariablen des EC2-Clusters von Amazon ECS fest

Festlegen Umgebungsvariablen des EC2-Clusters von Amazon ECS

1. Installieren Sie die Amazon ECS-CLI, falls noch nicht geschehen. Weitere Informationen finden Sie unter Installieren der Amazon-ECS-CLI.
2. Legen Sie den neuen Amazon-ECS-Clusternamen und die Region fest. Beispielsweise:

```
ECS_CLUSTER_NAME=ecs-ec2-memcached-tutorial
AWS_DEFAULT_REGION=ca-central-1
```

3. (Optional) Wenn Sie noch keinen Amazon ECS-Cluster mit dem EC2-Starttyp haben, auf dem Sie den Memcached-Beispiel-Workload und CloudWatch -Agenten installieren möchten, können Sie einen erstellen, indem Sie den folgenden Befehl eingeben.

```
ecs-cli up --capability-iam --size 1 \
--instance-type t3.medium \
--cluster $ECS_CLUSTER_NAME \
--region $AWS_REGION
```

Das erwartete Ergebnis dieses Befehls lautet wie folgt:

```
WARN[0000] You will not be able to SSH into your EC2 instances without a key pair.
INFO[0000] Using recommended Amazon Linux 2 AMI with ECS Agent 1.44.4 and Docker
 version 19.03.6-ce
INFO[0001] Created cluster                               cluster=ecs-ec2-memcached-
tutorial region=ca-central-1
INFO[0002] Waiting for your cluster resources to be created...
INFO[0002] Cloudformation stack status
 stackStatus=CREATE_IN_PROGRESS
INFO[0063] Cloudformation stack status
 stackStatus=CREATE_IN_PROGRESS
INFO[0124] Cloudformation stack status
 stackStatus=CREATE_IN_PROGRESS
VPC created: vpc-xxxxxxxxxxxxxxxxx
Security Group created: sg-xxxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxxx
Cluster creation succeeded.
```

Installieren der Memcached-Beispiel-Workload

So installieren Sie eine Memcached-Beispiel-Workload, die Prometheus-Metriken verfügbar macht

1. Laden Sie die AWS CloudFormation Memcached-Vorlage herunter, indem Sie den folgenden Befehl eingeben.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
cwagent-prometheus/sample_traffic/memcached/memcached-traffic-sample.yaml
```

2. Legen Sie die für Memcached zu erstellenden IAM-Rollennamen fest, indem Sie die folgenden Befehle eingeben.

```
MEMCACHED_ECS_TASK_ROLE_NAME=memcached-prometheus-demo-ecs-task-role-name
MEMCACHED_ECS_EXECUTION_ROLE_NAME=memcached-prometheus-demo-ecs-execution-role-name
```

3. Installieren Sie die Memcached-Beispiel-Workload, indem Sie den folgenden Befehl eingeben. Dieses Beispiel installiert die Workload im `host`-Netzwerkmodus.

```
MEMCACHED_ECS_NETWORK_MODE=host

aws cloudformation create-stack --stack-name Memcached-Prometheus-Demo-ECS-
$ECS_CLUSTER_NAME-EC2-$MEMCACHED_ECS_NETWORK_MODE \
    --template-body file://memcached-traffic-sample.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=ECSNetworkMode,ParameterValue=
$MEMCACHED_ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=
$MEMCACHED_ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$MEMCACHED_ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

Der AWS CloudFormation Stapel erstellt vier Ressourcen:

- Eine ECS-Aufgabenrolle
- Eine ECS-Aufgabenausführungsrolle
- Eine Memcached-Aufgabendefinition
- Ein Memcached-Service

In der Memcached-Aufgabendefinition werden zwei Container definiert:

- Der primäre Container führt eine einfache Memcached-Anwendung aus und öffnet Port 11211 für den Zugriff.
- Der andere Container führt den Redis-Exportprozess aus, um die Prometheus-Metriken auf Port 9150 verfügbar zu machen. Dies ist der Container, der vom Agenten entdeckt und gescrapt werden muss. CloudWatch

Konfigurieren Sie den CloudWatch Agenten für das Scraping von Memcached Prometheus-Metriken

So konfigurieren Sie den CloudWatch Agenten für das Scrapen von Memcached Prometheus-Metriken

1. Laden Sie die aktuelle Version der `cwagent-ecs-prometheus-metric-for-awsvpc.yaml` herunter, indem Sie einen der folgenden Befehle eingeben.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-awsvpc.yaml
   ```

2. Öffnen Sie die Datei mit einem Texteditor und finden Sie die vollständige CloudWatch Agentenkonfiguration hinter dem Schlüssel im `value` Abschnitt. `resource:CWAgentConfigSSMParameter`

   Fügen Sie dann im `ecs_service_discovery`-Abschnitt die folgende Konfiguration zum `task_definition_list`-Abschnitt hinzu.

   ```
   {
       "sd_job_name": "ecs-memcached",
       "sd_metrics_ports": "9150",
       "sd_task_definition_arn_pattern": ".*:task-definition/memcached-prometheus-demo.*:[0-9]+"
   },
   ```

   Für den `metric_declaration`-Abschnitt lässt die Standardeinstellung keine Memcached-Metriken zu. Fügen Sie den folgenden Abschnitt hinzu, um Memcached-Metriken zuzulassen. Beachten Sie unbedingt das vorhandene Einrückungsmuster.

   ```
   {
     "source_labels": ["container_name"],
     "label_matcher": "memcached-exporter-.*",
     "dimensions": [["ClusterName", "TaskDefinitionFamily"]],
     "metric_selectors": [
       "^memcached_current_(bytes|items|connections)$",
       "^memcached_items_(reclaimed|evicted)_total$",
       "^memcached_(written|read)_bytes_total$",
       "^memcached_limit_bytes$",
       "^memcached_commands_total$"
   ```

```
  ]
},
{
  "source_labels": ["container_name"],
  "label_matcher": "memcached-exporter-.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily","status","command"],
 ["ClusterName", "TaskDefinitionFamily","command"]],
  "metric_selectors": [
    "^memcached_commands_total$"
  ]
},
```

3. Wenn Sie den CloudWatch Agenten bereits im Amazon ECS-Cluster von bereitgestellt haben AWS CloudFormation, können Sie einen Änderungssatz erstellen, indem Sie die folgenden Befehle eingeben.

```
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
    --change-set-name memcached-scraping-support
```

4. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
5. Überprüfen des neu erstellten Änderungssatzes `memcached-scraping-support`. Sie sollten sehen, dass eine Änderung auf die `CWAgentConfigSSMParameter`-Ressource angewendet wurde. Führen Sie das Changeset aus und starten Sie die CloudWatch Agententask neu, indem Sie die folgenden Befehle eingeben.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
```

```
--desired-count 0 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

6. Warten Sie etwa 10 Sekunden und geben Sie den folgenden Befehl ein.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

7. Wenn Sie den CloudWatch Agenten mit der Erfassung von Prometheus-Metriken für den Cluster zum ersten Mal installieren, geben Sie bitte die folgenden Befehle ein:

```
ECS_NETWORK_MODEE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

Anzeigen Ihrer Memcached-Metriken

Dieses Tutorial sendet die folgenden Metriken an den ECS/ ContainerInsights /Prometheus-Namespace in. CloudWatch Sie können die CloudWatch Konsole verwenden, um die Metriken in diesem Namespace zu sehen.

Metrikname	Dimensionen
memcached_current_items	ClusterName , TaskDefinitionFamily
memcached_current_connections	ClusterName , TaskDefinitionFamily
memcached_limit_bytes	ClusterName , TaskDefinitionFamily
memcached_current_bytes	ClusterName , TaskDefinitionFamily
memcached_written_bytes_total	ClusterName , TaskDefinitionFamily
memcached_read_bytes_total	ClusterName , TaskDefinitionFamily
memcached_items_evicted_total	ClusterName , TaskDefinitionFamily
memcached_items_reclaimed_total	ClusterName , TaskDefinitionFamily
memcached_commands_total	ClusterName , TaskDefinitionFamily ClusterName , TaskDefinitionFamily, Befehl ClusterName , TaskDefinitionFamily, Status, Befehl

> **Note**
>
> Die Werte der command-Dimension können `delete`, `get`, `cas`, `set`, `decr`, `touch`, `incr`, oder `flush` sein.
> Die Werte der -Dimension können , `hit`, `miss`, oder `badval` sein.

Sie können auch ein CloudWatch Dashboard für Ihre Memcached Prometheus-Metriken erstellen.

So erstellen Sie ein Dashboard für Memcached-Prometheus-Metriken

1. Erstellen Sie Umgebungsvariablen und ersetzen Sie die folgenden Werte entsprechend Ihrer Bereitstellung.

```
DASHBOARD_NAME=your_memcached_cw_dashboard_name
ECS_TASK_DEF_FAMILY=memcached-prometheus-demo-$ECS_CLUSTER_NAME-EC2-
$MEMCACHED_ECS_NETWORK_MOD
```

2. Verwenden Sie den folgenden Befehl, um das Dashboard zu erstellen.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/ecs-task-definition-templates/deployment-mode/
replica-service/cwagent-prometheus/sample_cloudwatch_dashboards/memcached/
cw_dashboard_memcached.json \
| sed "s/{{YOUR_AWS_REGION}}/$AWS_REGION/g" \
| sed "s/{{YOUR_CLUSTER_NAME}}/$ECS_CLUSTER_NAME/g" \
| sed "s/{{YOUR_TASK_DEF_FAMILY}}/$ECS_TASK_DEF_FAMILY/g" \
| xargs -0 aws cloudwatch put-dashboard --dashboard-name ${DASHBOARD_NAME} --region
 $AWS_REGION --dashboard-body
```

Tutorial zum Scraping von Redis-Prometheus-Metriken auf Amazon ECS Fargate

Dieses Tutorial bietet eine praktische Einführung zum Scraping der Prometheus-Metriken einer Redis-Beispielanwendung in einem Amazon-ECS-Fargate-Cluster. Das Redis Prometheus-Exportziel wird vom CloudWatch Agenten mit Prometheus-Metrikunterstützung auf der Grundlage der Docker-Labels des Containers automatisch erkannt.

Redis (https://redis.io/) ist ein Open Source (BSD lizenziert), In-Memory-Datenstrukturspeicher, der als Datenbank, Cache und Message Broker verwendet wird. Weitere Informationen finden Sie unter redis.

redis_exporter (mit MIT-Lizenz lizenziert) wird verwendet, um die Redis-Prometheus-Metriken auf dem angegebenen Port verfügbar zu machen (Standard: 0.0.0.0:9121). Weitere Informationen finden Sie unter redis_exporter.

Die Docker-Images in den folgenden zwei Docker Hub-Repositories werden in diesem Tutorial verwendet:

- redis
- redis_exporter

Voraussetzung

Um Metriken aus einem Prometheus-Beispiel-Workload für Amazon ECS zu erfassen, müssen Sie Container Insights im Cluster ausführen. Informationen zur Installation von Container Insights finden Sie unter Einrichten von Container Insights für Amazon ECS.

Themen

- Legen Sie die Umgebungsvariable des Amazon-ECS-Fargate-Clusters fest
- Legen Sie die Netzwerkumgebungsvariablen für den Amazon-ECS-Fargate-Cluster fest
- Installieren der Redis-Beispiel-Workload
- Konfigurieren Sie den CloudWatch Agenten für das Scraping von Redis Prometheus-Metriken
- Anzeigen Ihrer Redis-Metriken

Legen Sie die Umgebungsvariable des Amazon-ECS-Fargate-Clusters fest

Festlegen der Umgebungsvariable des Amazon-ECS-Fargate-Clusters

1. Installieren Sie die Amazon ECS-CLI, falls noch nicht geschehen. Weitere Informationen finden Sie unter Installieren der Amazon-ECS-CLI.
2. Legen Sie den neuen Amazon-ECS-Clusternamen und die Region fest. Beispielsweise:

   ```
   ECS_CLUSTER_NAME=ecs-fargate-redis-tutorial
   AWS_DEFAULT_REGION=ca-central-1
   ```

3. (Optional) Wenn Sie noch keinen Amazon ECS Fargate-Cluster haben, auf dem Sie den Redis-Beispiel-Workload und CloudWatch -Agenten installieren möchten, können Sie einen erstellen, indem Sie den folgenden Befehl eingeben.

```
ecs-cli up --capability-iam \
--cluster $ECS_CLUSTER_NAME \
--launch-type FARGATE \
--region $AWS_DEFAULT_REGION
```

Das erwartete Ergebnis dieses Befehls lautet wie folgt:

```
INFO[0000] Created cluster   cluster=ecs-fargate-redis-tutorial region=ca-central-1
INFO[0001] Waiting for your cluster resources to be created...
INFO[0001] Cloudformation stack status   stackStatus=CREATE_IN_PROGRESS
VPC created: vpc-xxxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxxx
Cluster creation succeeded.
```

Legen Sie die Netzwerkumgebungsvariablen für den Amazon-ECS-Fargate-Cluster fest

Festlegen der Netzwerkumgebungsvariablen für den Amazon-ECS-Fargate-Cluster

1. Legen Sie die VPC- und Subnetz-ID des Amazon-ECS-Clusters fest. Wenn Sie im vorherigen Verfahren einen neuen Cluster erstellt haben, werden diese Werte im Ergebnis des endgültigen Befehls angezeigt. Andernfalls verwenden Sie die IDs des vorhandenen Clusters, den Sie mit Redis verwenden möchten.

   ```
   ECS_CLUSTER_VPC=vpc-xxxxxxxxxxxxxxxxx
   ECS_CLUSTER_SUBNET_1=subnet-xxxxxxxxxxxxxxxxx
   ECS_CLUSTER_SUBNET_2=subnet-xxxxxxxxxxxxxxxxx
   ```

2. In diesem Tutorial werden wir die Redis-Anwendung und den CloudWatch Agenten in der Standardsicherheitsgruppe der VPC des Amazon ECS-Clusters installieren. Die Standardsicherheitsgruppe erlaubt alle Netzwerkverbindungen innerhalb derselben Sicherheitsgruppe, sodass der CloudWatch Agent die in den Redis-Containern offengelegten Prometheus-Metriken auslesen kann. In einer echten Produktionsumgebung möchten Sie möglicherweise spezielle Sicherheitsgruppen für die Redis-Anwendung und den CloudWatch Redis-Agenten erstellen und benutzerdefinierte Berechtigungen für diese einrichten.

   Geben Sie den folgenden Befehl ein, um die Standard-Sicherheitsgruppen-ID abzurufen.

```
aws ec2 describe-security-groups \
--filters Name=vpc-id,Values=$ECS_CLUSTER_VPC  \
--region $AWS_DEFAULT_REGION
```

Legen Sie dann die Standardsicherheitsgruppenvariable des Fargate-Clusters fest, indem Sie den folgenden Befehl eingeben und ihn durch den Wert *my-default-security-group*ersetzen, den Sie im vorherigen Befehl gefunden haben.

```
ECS_CLUSTER_SECURITY_GROUP=my-default-security-group
```

Installieren der Redis-Beispiel-Workload

So installieren Sie eine Redis-Beispiel-Workload, die Prometheus-Metriken verfügbar macht

1. Laden Sie die AWS CloudFormation Redis-Vorlage herunter, indem Sie den folgenden Befehl eingeben.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
cwagent-prometheus/sample_traffic/redis/redis-traffic-sample.yaml
```

2. Legen Sie die für Redis zu erstellenden IAM-Rollennamen fest, indem Sie die folgenden Befehle eingeben.

```
REDIS_ECS_TASK_ROLE_NAME=redis-prometheus-demo-ecs-task-role-name
REDIS_ECS_EXECUTION_ROLE_NAME=redis-prometheus-demo-ecs-execution-role-name
```

3. Installieren Sie die Redis-Beispiel-Workload, indem Sie den folgenden Befehl eingeben.

```
aws cloudformation create-stack --stack-name Redis-Prometheus-Demo-ECS-
$ECS_CLUSTER_NAME-fargate-awsvpc \
    --template-body file://redis-traffic-sample.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=SecurityGroupID,ParameterValue=
$ECS_CLUSTER_SECURITY_GROUP \
                 ParameterKey=SubnetID,ParameterValue=$ECS_CLUSTER_SUBNET_1 \
                 ParameterKey=TaskRoleName,ParameterValue=$REDIS_ECS_TASK_ROLE_NAME
 \
```

```
                    ParameterKey=ExecutionRoleName,ParameterValue=
$REDIS_ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_DEFAULT_REGION
```

Der AWS CloudFormation Stack erstellt vier Ressourcen:

- Eine ECS-Aufgabenrolle
- Eine ECS-Aufgabenausführungsrolle
- Eine Redis-Aufgabendefinition
- Ein Redis-Service

In der Redis-Aufgabendefinition werden zwei Container definiert:

- Der primäre Container führt eine einfache Redis-Anwendung aus und öffnet Port 6379 für den Zugriff.
- Der andere Container führt den Redis-Exportprozess aus, um die Prometheus-Metriken auf Port 9121 verfügbar zu machen. Dies ist der Container, der vom Agenten entdeckt und gescrapt werden muss. CloudWatch Das folgende Docker-Label ist so definiert, dass der CloudWatch Agent diesen Container anhand dessen erkennen kann.

```
ECS_PROMETHEUS_EXPORTER_PORT: 9121
```

Konfigurieren Sie den CloudWatch Agenten für das Scraping von Redis Prometheus-Metriken

So konfigurieren Sie den CloudWatch Agenten für das Scrapen von Redis Prometheus-Metriken

1. Laden Sie die aktuelle Version der `cwagent-ecs-prometheus-metric-for-awsvpc.yaml` herunter, indem Sie einen der folgenden Befehle eingeben.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-
awsvpc.yaml
```

2. Öffnen Sie die Datei mit einem Texteditor und finden Sie die vollständige CloudWatch Agentenkonfiguration hinter dem `value` Schlüssel im Abschnitt. `resource:CWAgentConfigSSMParameter`

   Dann wird im hier gezeigten Abschnitt `ecs_service_discovery` die `docker_label`-basierte Serviceerkennung mit den auf `ECS_PROMETHEUS_EXPORTER_PORT` basierenden Standardeinstellungen aktiviert, die der Docker-Bezeichnung entsprechen, das wir in der Redis-ECS-Aufgabendefinition definiert haben. So müssen wir keine Änderungen in diesem Abschnitt vornehmen:

   ```
   ecs_service_discovery": {
     "sd_frequency": "1m",
     "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
   *  "docker_label": {
     },*
     ...
   ```

   Für den `metric_declaration`-Abschnitt lässt die Standardeinstellung keine Redis-Metriken zu. Fügen Sie den folgenden Abschnitt hinzu, um Redis-Metriken zuzulassen. Beachten Sie unbedingt das vorhandene Einrückungsmuster.

   ```
   {
     "source_labels": ["container_name"],
     "label_matcher": "^redis-exporter-.*$",
     "dimensions": [["ClusterName","TaskDefinitionFamily"]],
     "metric_selectors": [
       "^redis_net_(in|out)put_bytes_total$",
       "^redis_(expired|evicted)_keys_total$",
       "^redis_keyspace_(hits|misses)_total$",
       "^redis_memory_used_bytes$",
       "^redis_connected_clients$"
     ]
   },
   {
     "source_labels": ["container_name"],
     "label_matcher": "^redis-exporter-.*$",
     "dimensions": [["ClusterName","TaskDefinitionFamily","cmd"]],
     "metric_selectors": [
       "^redis_commands_total$"
     ]
   },
   ```

```
{
  "source_labels": ["container_name"],
  "label_matcher": "^redis-exporter-.*$",
  "dimensions": [["ClusterName","TaskDefinitionFamily","db"]],
  "metric_selectors": [
    "^redis_db_keys$"
  ]
},
```

3. Wenn Sie den CloudWatch Agenten bereits im Amazon ECS-Cluster von bereitgestellt haben AWS CloudFormation, können Sie einen Änderungssatz erstellen, indem Sie die folgenden Befehle eingeben.

```
ECS_LAUNCH_TYPE=FARGATE
CREATE_IAM_ROLES=True
ECS_CLUSTER_SUBNET=$ECS_CLUSTER_SUBNET_1
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
$ECS_CLUSTER_NAME-$ECS_LAUNCH_TYPE-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSLaunchType,ParameterValue=$ECS_LAUNCH_TYPE \
                 ParameterKey=SecurityGroupID,ParameterValue=
$ECS_CLUSTER_SECURITY_GROUP \
                 ParameterKey=SubnetID,ParameterValue=$ECS_CLUSTER_SUBNET \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --change-set-name redis-scraping-support
```

4. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
5. Überprüfen des neu erstellten Änderungssatzes `redis-scraping-support`. Sie sollten sehen, dass eine Änderung auf die `CWAgentConfigSSMParameter`-Ressource angewendet wurde. Führen Sie das Changeset aus und starten Sie die CloudWatch Agententask neu, indem Sie die folgenden Befehle eingeben.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service cwagent-prometheus-replica-service-$ECS_LAUNCH_TYPE-awsvpc \
--region ${AWS_DEFAULT_REGION}
```

6. Warten Sie etwa 10 Sekunden und geben Sie den folgenden Befehl ein.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-$ECS_LAUNCH_TYPE-awsvpc \
--region ${AWS_DEFAULT_REGION}
```

7. Wenn Sie den CloudWatch Agenten mit der Erfassung von Prometheus-Metriken für den Cluster zum ersten Mal installieren, geben Sie bitte die folgenden Befehle ein:

```
ECS_LAUNCH_TYPE=FARGATE
CREATE_IAM_ROLES=True
ECS_CLUSTER_SUBNET=$ECS_CLUSTER_SUBNET_1
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
$ECS_CLUSTER_NAME-$ECS_LAUNCH_TYPE-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSLaunchType,ParameterValue=$ECS_LAUNCH_TYPE \
                 ParameterKey=SecurityGroupID,ParameterValue=
$ECS_CLUSTER_SECURITY_GROUP \
                 ParameterKey=SubnetID,ParameterValue=$ECS_CLUSTER_SUBNET \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION}
```

Anzeigen Ihrer Redis-Metriken

Dieses Tutorial sendet die folgenden Metriken an den ECS/ ContainerInsights /Prometheus-Namespace in. CloudWatch Sie können die CloudWatch Konsole verwenden, um die Metriken in diesem Namespace zu sehen.

Metrikname	Dimensionen	
`redis_net_input_bytes_total`	ClusterName, `TaskDefinitionFamily`	
`redis_net_output_bytes_total`	ClusterName, `TaskDefinitionFamily`	
`redis_expired_keys_total`	ClusterName, `TaskDefinitionFamily`	
`redis_evicted_keys_total`	ClusterName, `TaskDefinitionFamily`	
`redis_keyspace_hits_total`	ClusterName, `TaskDefinitionFamily`	
`redis_keyspace_misses_total`	ClusterName, `TaskDefinitionFamily`	
`redis_memory_used_bytes`	ClusterName, `TaskDefinitionFamily`	
`redis_connected_clients`	ClusterName, `TaskDefinitionFamily`	

Metrikname	Dimensionen	
redis_commands_total	ClusterName , TaskDefinitionFamily , cmd	
redis_db_keys	ClusterName , TaskDefinitionFamily , db	

> **Note**
>
> Die Werte der cmd-Dimension können append, client, command, config, dbsize, flushall, get, incr, info, latency oder slowlog sein.
> Die Werte der Db-Dimension können db0 oder db15 sein.

Sie können auch ein CloudWatch Dashboard für Ihre Redis Prometheus-Metriken erstellen.

So erstellen Sie ein Dashboard für Redis-Prometheus-Metriken

1. Erstellen Sie Umgebungsvariablen und ersetzen Sie die folgenden Werte entsprechend Ihrer Bereitstellung.

```
DASHBOARD_NAME=your_cw_dashboard_name
ECS_TASK_DEF_FAMILY=redis-prometheus-demo-$ECS_CLUSTER_NAME-fargate-awsvpc
```

2. Verwenden Sie den folgenden Befehl, um das Dashboard zu erstellen.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_cloudwatch_dashboards/redis/cw_dashboard_redis.json \
| sed "s/{{YOUR_AWS_REGION}}/${REGION_NAME}/g" \
| sed "s/{{YOUR_CLUSTER_NAME}}/${CLUSTER_NAME}/g" \
| sed "s/{{YOUR_NAMESPACE}}/${NAMESPACE}/g" \
```

Einrichten und Konfigurieren der Prometheus-Metrikensammlung in Amazon EKS oder Kubernetes-Clustern

Um Prometheus-Metriken von Clustern zu sammeln, auf denen Amazon EKS oder Kubernetes ausgeführt werden, können Sie den CloudWatch Agenten als Collector oder die AWS Distro for Collector verwenden. OpenTelemetry Informationen zur Verwendung von AWS Distro for Collector finden Sie unter https://aws-otel.github.io/docs/getting-started/container-insights/eks-prometheus. OpenTelemetry

In den folgenden Abschnitten wird erklärt, wie Sie Prometheus-Metriken mithilfe des CloudWatch Agenten sammeln. Sie erklären, wie der CloudWatch Agent mit Prometheus-Überwachung auf Clustern installiert wird, auf denen Amazon EKS oder Kubernetes ausgeführt wird, und wie der Agent so konfiguriert wird, dass er zusätzliche Ziele scrapt. Sie enthalten auch optionale Tutorials zum Einrichten von Beispiel-Workloads zum Testen mit der Prometheus-Überwachung.

Themen

Installieren Sie den CloudWatch Agenten mit der Erfassung von Prometheus-Metriken auf Amazon EKS- und Kubernetes-Clustern

In diesem Abschnitt wird erklärt, wie Sie den CloudWatch Agenten mit Prometheus-Überwachung in einem Cluster einrichten, auf dem Amazon EKS oder Kubernetes ausgeführt wird. Danach scrapt und importiert der Agent automatisch Metriken für die folgenden Workloads, die in diesem Cluster ausgeführt werden.

- AWS App Mesh
- NGINX
- Memcached
- Java/JMX
- HAProxy
- Fluent Bit

Sie können den Agenten auch so konfigurieren, dass er aus weiteren Prometheus-Workloads und -Quellen importiert.

Bevor Sie diese Schritte ausführen, um den CloudWatch Agenten für die Prometheus-Metrikerfassung zu installieren, müssen Sie einen Cluster auf Amazon EKS oder einen Kubernetes-Cluster auf einer Amazon EC2 EC2-Instance ausführen.

Anforderungen an VPC-Sicherheitsgruppen

Die Eingangsregeln der Sicherheitsgruppen für die Prometheus-Workloads müssen die Prometheus-Ports für den CloudWatch Agenten öffnen, damit er die Prometheus-Metriken über die private IP scrapen kann.

Die Ausgangsregeln der Sicherheitsgruppe für den CloudWatch Agenten müssen es dem CloudWatch Agenten ermöglichen, über eine private IP eine Verbindung zum Port der Prometheus-Workloads herzustellen.

Themen

Installieren Sie den CloudWatch Agenten mit der Erfassung von Prometheus-Metriken auf Amazon EKS- und Kubernetes-Clustern

In diesem Abschnitt wird erklärt, wie Sie den CloudWatch Agenten mit Prometheus-Überwachung in einem Cluster einrichten, auf dem Amazon EKS oder Kubernetes ausgeführt wird. Danach scrapt und importiert der Agent automatisch Metriken für die folgenden Workloads, die in diesem Cluster ausgeführt werden.

- AWS App Mesh
- NGINX
- Memcached
- Java/JMX
- HAProxy
- Fluent Bit

Sie können den Agenten auch so konfigurieren, dass er aus weiteren Prometheus-Workloads und -Quellen importiert.

Bevor Sie diese Schritte ausführen, um den CloudWatch Agenten für die Prometheus-Metrikerfassung zu installieren, müssen Sie einen Cluster auf Amazon EKS oder einen Kubernetes-Cluster auf einer Amazon EC2 EC2-Instance ausführen.

Anforderungen an VPC-Sicherheitsgruppen

Die Eingangsregeln der Sicherheitsgruppen für die Prometheus-Workloads müssen die Prometheus-Ports für den CloudWatch Agenten öffnen, damit er die Prometheus-Metriken über die private IP scrapen kann.

Die Ausgangsregeln der Sicherheitsgruppe für den CloudWatch Agenten müssen es dem CloudWatch Agenten ermöglichen, über eine private IP eine Verbindung zum Port der Prometheus-Workloads herzustellen.

Themen

- Einrichten von IAM-Rollen
- Installation des CloudWatch Agenten zur Erfassung von Prometheus-Metriken

Einrichten von IAM-Rollen

Der erste Schritt besteht darin, die erforderliche IAM-Richtlinie in dem Cluster einzurichten. Es gibt zwei Methoden:

- Richten Sie eine IAM-Rolle für ein Servicekonto ein, die auch als Servicerolle bezeichnet wird. Diese Methode funktioniert sowohl für den EC2-Starttyp als auch für den Fargate-Starttyp.
- Fügen Sie der IAM-Rolle eine IAM-Richtlinie hinzu, die für den Cluster verwendet wird. Dies funktioniert nur für den Starttyp EC2.

Einrichten einer Servicerolle (EC2-Starttyp und Fargate-Starttyp)

Geben Sie zum Einrichten einer Service-Rolle den folgenden Befehl ein. Ersetzen Sie *MyCluster* durch den Namen des Clusters.

```
eksctl create iamserviceaccount \
 --name cwagent-prometheus \
```

```
--namespace amazon-cloudwatch \
 --cluster MyCluster \
--attach-policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy \
--approve \
--override-existing-serviceaccounts
```

Hinzufügen einer Richtlinie zur IAM-Rolle des Clusters (nur EC2-Starttyp)

So richten Sie die IAM-Richtlinie in einem Cluster für die Prometheus-Unterstützung ein:

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Sie müssen das Präfix des IAM-Rollennamens für den Cluster finden. Markieren Sie dazu das Kontrollkästchen neben dem Namen einer Instance, die sich im Cluster befindet, und wählen Sie Actions (Aktionen), Instance Settings (Instance-Einstellungen), Attach/Replace IAM Role (Anfügen/Ersetzen einer IAM-Rolle). Kopieren Sie dann das Präfix der IAM-Rolle, z. B. `eksctl-dev303-workshop-nodegroup`.
4. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
5. Wählen Sie im Navigationsbereich Rollen aus.
6. Verwenden Sie das Suchfeld, um das Präfix zu finden, das Sie zuvor in diesem Verfahren kopiert haben, und wählen Sie diese Rolle aus.
7. Wählen Sie Richtlinien anfügen.
8. Verwenden Sie das Suchfeld, um zu suchen CloudWatchAgentServerPolicy. Aktivieren Sie das Kontrollkästchen neben CloudWatchAgentServerPolicyund wählen Sie Richtlinie anhängen aus.

Installation des CloudWatch Agenten zur Erfassung von Prometheus-Metriken

Sie müssen den CloudWatch Agenten im Cluster installieren, um die Metriken zu sammeln. Die Installation des Agenten unterscheidet sich für Amazon-EKS-Cluster und Kubernetes-Cluster.

Löschen Sie frühere Versionen des CloudWatch Agenten mit Prometheus-Unterstützung

Wenn Sie bereits eine Version des CloudWatch Agenten mit Prometheus-Unterstützung in Ihrem Cluster installiert haben, müssen Sie diese Version löschen, indem Sie den folgenden Befehl eingeben. Dies ist nur für frühere Versionen des Agenten mit Prometheus-Unterstützung erforderlich. Sie müssen den CloudWatch Agenten, der Container Insights ohne Prometheus-Unterstützung aktiviert, nicht löschen.

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
```

Installation des CloudWatch Agenten auf Amazon EKS-Clustern mit dem EC2-Starttyp

Gehen Sie folgendermaßen vor, um den CloudWatch Agenten mit Prometheus-Unterstützung auf einem Amazon EKS-Cluster zu installieren.

Um den CloudWatch Agenten mit Prometheus-Unterstützung auf einem Amazon EKS-Cluster zu installieren

1. Geben Sie den folgenden Befehl ein, um zu prüfen, ob der `amazon-cloudwatch`-Namespace bereits erstellt wurde:

   ```
   kubectl get namespace
   ```

2. Wenn `amazon-cloudwatch` nicht in den Ergebnissen angezeigt wird, erstellen Sie ihn, indem Sie den folgenden Befehl eingeben:

   ```
   kubectl create namespace amazon-cloudwatch
   ```

3. Geben Sie den folgenden Befehl ein, um den Agenten mit der Standardkonfiguration bereitzustellen und Daten an die AWS Region senden zu lassen, in der er installiert ist:

   ```
   kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
   container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
   service/cwagent-prometheus/prometheus-eks.yaml
   ```

   Gehen Sie folgendermaßen vor, damit der Agent stattdessen Daten an eine andere Region sendet:

   a. Laden Sie die YAML-Datei für den Agenten herunter, indem Sie den folgenden Befehl eingeben:

      ```
      curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
      container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
      service/cwagent-prometheus/prometheus-eks.yaml
      ```

   b. Öffnen Sie die Datei mit einem Texteditor und suchen Sie nach dem `cwagentconfig.json`-Block der Datei.

   c. Fügen Sie die markierten Zeilen hinzu und geben Sie die gewünschte Region an:

```
cwagentconfig.json: |
    {
      "agent": {
        "region": "us-east-2"
      },
      "logs": { ...
```

d. Speichern Sie die Datei und stellen Sie den Agenten mithilfe der aktualisierten Datei bereit.

```
kubectl apply -f prometheus-eks.yaml
```

Installation des CloudWatch Agenten auf Amazon EKS-Clustern mit dem Starttyp Fargate

Gehen Sie wie folgt vor, um den CloudWatch Agenten mit Prometheus-Unterstützung auf einem Amazon EKS-Cluster mit dem Starttyp Fargate zu installieren.

Um den CloudWatch Agenten mit Prometheus-Unterstützung auf einem Amazon EKS-Cluster mit dem Starttyp Fargate zu installieren

1. Geben Sie den folgenden Befehl ein, um ein Fargate-Profil für den CloudWatch Agenten zu erstellen, damit er innerhalb des Clusters ausgeführt werden kann. *MyCluster*Ersetzen Sie es durch den Namen des Clusters.

   ```
   eksctl create fargateprofile --cluster MyCluster \
   --name amazon-cloudwatch \
   --namespace amazon-cloudwatch
   ```

2. Geben Sie den folgenden Befehl ein, um den CloudWatch Agenten zu installieren. *MyCluster*Ersetzen Sie ihn durch den Namen des Clusters. Dieser Name wird im Protokollgruppennamen, in dem die vom Agenten erfassten Protokollereignisse gespeichert werden, und auch als Dimension für die vom Agenten erfassten Metriken verwendet.

   Ersetzen Sie die *Region* durch den Namen der Region, in die die Metriken gesendet werden sollen. z. B. `us-west-1`.

   ```
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
   insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
   prometheus/prometheus-eks-fargate.yaml |
   sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" |
   ```

```
kubectl apply -f -
```

Den CloudWatch Agenten auf einem Kubernetes-Cluster installieren

Um den CloudWatch Agenten mit Prometheus-Unterstützung auf einem Cluster zu installieren, auf dem Kubernetes ausgeführt wird, geben Sie den folgenden Befehl ein:

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/prometheus-k8s.yaml |
sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" |
kubectl apply -f -
```

Ersetzen Sie ihn durch *`MyCluster`*den Namen des Clusters. Dieser Name wird im Protokollgruppennamen, in dem die vom Agenten erfassten Protokollereignisse gespeichert werden, und auch als Dimension für die vom Agenten erfassten Metriken verwendet.

Ersetzen Sie *`Region`* durch den Namen der AWS Region, in die die Metriken gesendet werden sollen. z. B. **`us-west-1`**.

Überprüfen, ob der Agent ausgeführt wird

Auf Amazon-EKS- und Kubernetes-Clustern können Sie den folgenden Befehl eingeben, um zu prüfen, ob der Agent ausgeführt wird.

```
kubectl get pod -l "app=cwagent-prometheus" -n amazon-cloudwatch
```

Wenn die Ergebnisse einen einzelnen CloudWatch Agent-Pod im `Running` Status enthalten, läuft der Agent und sammelt Prometheus-Metriken. Standardmäßig sammelt der CloudWatch Agent jede Minute Metriken für App Mesh, NGINX, Memcached, Java/JMX und HAProxy. Weitere Informationen zu diesen Metriken finden Sie unter Vom Agenten gesammelte Prometheus-Metriken CloudWatch . Eine Anleitung, wie Sie Ihre Prometheus-Metriken einsehen können, finden Sie unter CloudWatch Anzeigen Ihrer Prometheus-Metriken

Sie können den CloudWatch Agenten auch so konfigurieren, dass er Metriken von anderen Prometheus-Exporteuren sammelt. Weitere Informationen finden Sie unter Scraping zusätzlicher Prometheus-Quellen und Importieren dieser Metriken.

Scraping zusätzlicher Prometheus-Quellen und Importieren dieser Metriken

Der CloudWatch Agent mit Prometheus-Überwachung benötigt zwei Konfigurationen, um die Prometheus-Metriken zu erfassen. Er folgt der standardmäßigen Prometheus-Konfiguration, wie in <scrape_config> in der Prometheus-Dokumentation erläutert. Die andere ist für die Agentenkonfiguration vorgesehen. CloudWatch

Für Amazon-EKS-Cluster sind die Konfigurationen in `prometheus-eks.yaml` (für den Starttyp EC2) oder `prometheus-eks-fargate.yaml` (für den Starttyp Fargate) als zwei Konfigurationszuordnungen definiert:

- Der `name: prometheus-config`-Abschnitt enthält die Einstellungen für Prometheus-Scraping.
- Der `name: prometheus-cwagentconfig` Abschnitt enthält die Konfiguration für den CloudWatch Agenten. In diesem Abschnitt können Sie konfigurieren, wie die Prometheus-Metriken von erfasst werden. CloudWatch Sie geben beispielsweise an, in welche Metriken importiert werden sollen CloudWatch, und definieren deren Dimensionen.

Für Kubernetes-Cluster, die auf Amazon-EC2-Instances ausgeführt werden, sind die Konfigurationen in der `prometheus-k8s.yaml` YAML-Datei als zwei Konfigurationszuordnungen definiert:

- Der `name: prometheus-config`-Abschnitt enthält die Einstellungen für Prometheus-Scraping.
- Der `name: prometheus-cwagentconfig` Abschnitt enthält die Konfiguration für den CloudWatch Agenten.

Um zusätzliche Prometheus-Metrikquellen zu scrapen und diese Metriken zu importieren CloudWatch, ändern Sie sowohl die Prometheus-Scrape-Konfiguration als auch die Agentenkonfiguration und stellen dann den CloudWatch Agenten mit der aktualisierten Konfiguration erneut bereit.

Anforderungen an VPC-Sicherheitsgruppen

Die Eingangsregeln der Sicherheitsgruppen für die Prometheus-Workloads müssen die Prometheus-Ports für den CloudWatch Agenten öffnen, damit er die Prometheus-Metriken über die private IP scrapen kann.

Die Ausgangsregeln der Sicherheitsgruppe für den CloudWatch Agenten müssen es dem CloudWatch Agenten ermöglichen, über eine private IP eine Verbindung zum Port der Prometheus-Workloads herzustellen.

Prometheus-Scrape-Konfiguration

Der CloudWatch Agent unterstützt die standardmäßigen Prometheus-Scrape-Konfigurationen, wiehttps://prometheus.io/docs/prometheus/latest/configuration/configuration/#scrape_config <scrape_config>in der Prometheus-Dokumentation dokumentiert. Sie können diesen Abschnitt bearbeiten, um die Konfigurationen zu aktualisieren, die sich bereits in dieser Datei befinden, und zusätzliche Prometheus-Scraping-Ziele hinzufügen. Standardmäßig enthält die Beispielkonfigurationsdatei die folgenden globalen Konfigurationszeilen:

```
global:
  scrape_interval: 1m
  scrape_timeout: 10s
```

- scrape_interval – Definiert, wie oft das Scraping von zielen durchgeführt werden soll.
- scrape_timeout – Definiert, wie lange gewartet werden soll, bis für eine Scrape-Anforderung eine Zeitüberschreitung eintritt.

Sie können auch verschiedene Werte für diese Einstellungen auf Auftragsebene definieren, um die globalen Konfigurationen zu überschreiben.

Prometheus-Scraping-Aufträge

Für die YAML-Dateien des CloudWatch Agenten sind bereits einige Standard-Scraping-Jobs konfiguriert. In `prometheus-eks.yaml` werden beispielsweise die Standard-Scraping-Aufträge in den `job_name`-Zeilen im Abschnitt `scrape_configs` konfiguriert. In dieser Datei kratzt der folgende Standard-`kubernetes-pod-jmx`-Abschnitt JMX-Exporter-Metriken.

```
   - job_name: 'kubernetes-pod-jmx'
      sample_limit: 10000
      metrics_path: /metrics
      kubernetes_sd_configs:
      - role: pod
      relabel_configs:
      - source_labels: [__address__]
        action: keep
        regex: '.*:9404$'
      - action: labelmap
        regex: __meta_kubernetes_pod_label_(.+)
      - action: replace
        source_labels:
```

```
        - __meta_kubernetes_namespace
          target_label: Namespace
        - source_labels: [__meta_kubernetes_pod_name]
          action: replace
          target_label: pod_name
        - action: replace
          source_labels:
          - __meta_kubernetes_pod_container_name
          target_label: container_name
        - action: replace
          source_labels:
          - __meta_kubernetes_pod_controller_name
          target_label: pod_controller_name
        - action: replace
          source_labels:
          - __meta_kubernetes_pod_controller_kind
          target_label: pod_controller_kind
        - action: replace
          source_labels:
          - __meta_kubernetes_pod_phase
          target_label: pod_phase
```

Jedes dieser Standardziele wird gelöscht, und die Metriken werden im eingebetteten Metrikformat CloudWatch an Protokollereignisse gesendet. Weitere Informationen finden Sie unter Einbetten von Metriken in Protokollen.

*Protokollereignisse von Amazon EKS- und Kubernetes-Clustern werden in der Protokollgruppe **/aws/containerinsights/ cluster_name /prometheus** in Logs gespeichert.* CloudWatch Protokollereignisse von Amazon-ECS-Clustern werden in der Protokollgruppe /aws/ecs/containerinsights/***cluster_name***/prometheus gespeichert.

Jeder Scraping-Auftrag ist in einem anderen Protokoll-Stream in dieser Protokollgruppe enthalten. Beispielsweise ist der Prometheus-Scraping-Auftrag `kubernetes-pod-appmesh-envoy` für App Mesh definiert. *Alle App Mesh Prometheus-Metriken von Amazon EKS- und Kubernetes-Clustern werden an den Protokollstream mit dem Namen **/aws/containerinsights/ cluster_name >prometheus//**gesendet.* kubernetes-pod-appmesh-envoy

Um ein neues Scraping-Ziel hinzuzufügen, fügen Sie dem Abschnitt `scrape_configs` der YAML-Datei einen neuen `job_name`-Abschnitt hinzu und starten Sie den Agenten neu. Ein Beispiel für

diesen Prozess finden Sie unter Tutorial zum Hinzufügen eines neuen Prometheus-Scrape-Ziels: Prometheus-API-Server-Metriken.

CloudWatch Agentenkonfiguration für Prometheus

Die CloudWatch Agentenkonfigurationsdatei enthält einen `prometheus` Abschnitt `metrics_collected` für die Prometheus-Scraping-Konfiguration. Es sind folgende Konfigurationsoptionen enthalten:

- Clustername – Gibt den Clusternamen an, der als Bezeichnung im Protokollereignis hinzugefügt werden soll. Dies ist ein optionales Feld. Wenn Sie es weglassen, kann der Agent den Amazon-EKS- oder Kubernetes-Clusternamen erkennen.
- log_group_name – Gibt den Namen der Protokollgruppe für die Prometheus-Scrape-Metriken an. Dies ist ein optionales Feld. Wenn Sie es weglassen, wird /aws/containerinsights/ ***cluster_name /prometheus für Protokolle von Amazon EKS- und Kubernetes-Clustern CloudWatch*** verwendet.
- prometheus_config_path – gibt den Pfad der Prometheus-Scrape-Konfigurationsdatei an. Wenn der Wert dieses Felds mit `env:` beginnt, wird der Inhalt der Prometheus-Scrape-Konfigurationsdatei aus der Umgebungsvariablen des Containers abgerufen. Ändern Sie dieses Feld nicht.
- ecs_service_discovery – ist der Abschnitt zum Angeben der Konfiguration für die Amazon-ECS-Prometheus-Serviceerkennung. Weitere Informationen finden Sie unter Ausführliche Anleitung zu Autodiscovery auf Amazon-ECS-Clustern.

  Der Abschnitt `ecs_service_discovery` kann die folgenden Felder enthalten:

  - `sd_frequency` ist die Häufigkeit, mit der die Prometheus-Exporteure entdeckt werden. Geben Sie eine Zahl und ein Einheitensuffix an. Zum Beispiel `1m` für einmal pro Minute oder `30s` für einmal pro 30 Sekunden. Gültige Einheitensuffixe sind `ns`, `us`, `ms`, `s`, `m` und `h`.

    Dies ist ein optionales Feld. Der Standardwert ist 60 Sekunden (1 Minute).
  - `sd_target_cluster` ist der Name des Amazon-ECS-Ziel-Clusters für die automatische Erkennung. Dies ist ein optionales Feld. Der Standard ist der Name des Amazon ECS-Clusters, auf dem der CloudWatch Agent installiert ist.
  - `sd_cluster_region` ist die Region des Amazon-ECS-Ziel-Clusters. Dies ist ein optionales Feld. Die Standardeinstellung ist die Region des Amazon ECS-Clusters, in der der CloudWatch Agent installiert ist.
  - `sd_result_file` ist der Pfad der YAML-Datei für die Prometheus Zielergebnisse. Die Prometheus-Scrape-Konfiguration bezieht sich auf diese Datei.

- `docker_label` ist ein optionaler Abschnitt, mit dem Sie die Konfiguration für die Docker-Beschriftungs-basierte Service-Discovery angeben können. Wenn Sie diesen Abschnitt auslassen, wird die Docker-Bezeichnungs-basierte Erkennung nicht verwendet. Dieser Abschnitt kann die folgenden Felder enthalten:
  - `sd_port_label` ist der Docker-Bezeichnungsname des Containers, der den Container-Port für Prometheus Metriken angibt. Der Standardwert ist `ECS_PROMETHEUS_EXPORTER_PORT`. Wenn der Container dieses Docker-Label nicht hat, überspringt der CloudWatch Agent es.
  - `sd_metrics_path_label` ist der Docker-Bezeichnungsname des Containers, der den Pfad für Prometheus Metriken angibt. Der Standardwert ist `ECS_PROMETHEUS_METRICS_PATH`. Wenn der Container nicht über diese Docker-Bezeichnung verfügt, nimmt der Agent den Standardpfad `/metrics` an.
  - `sd_job_name_label` ist der Docker-Bezeichnungsname des Containers, der den Container-Scraping-Auftrag-Namen für Prometheus angibt. Der Standardwert ist `job`. Wenn der Container dieses Docker-Label nicht hat, verwendet der CloudWatch Agent den Jobnamen in der Prometheus-Scrape-Konfiguration.
- `task_definition_list` ist ein optionaler Abschnitt, den Sie verwenden können, um die Konfiguration der aufgabendefinitionsbasierten Serviceerkennung anzugeben. Wenn Sie diesen Abschnitt auslassen, wird die aufgabendefinitionsbasierte Erkennung nicht verwendet. Dieser Abschnitt kann die folgenden Felder enthalten:
  - `sd_task_definition_arn_pattern` ist das Muster, das verwendet wird, um die zu erkennenden Amazon-ECS-Aufgabendefinitionen anzugeben. Dies ist ein regulärer Ausdruck.
  - `sd_metrics_ports` listet den containerPort für die Prometheus-Metriken auf. Trennen Sie die ContainerPorts durch Semikolons.
  - `sd_container_name_pattern` gibt die Namen des Amazon-ECS-Aufgabencontainers an. Dies ist ein regulärer Ausdruck.
  - `sd_metrics_path` gibt den Prometheus-Metrikpfad an. Wenn Sie dies weglassen, übernimmt der Agent den Standardpfad `/metrics`
  - `sd_job_name` gibt den Namen des Prometheus -Scrape-Auftrags an. Wenn Sie dieses Feld weglassen, verwendet der CloudWatch Agent den Jobnamen in der Prometheus-Scrape-Konfiguration.
- metric_declaration – sind Abschnitte, die das Array von Protokollen mit eingebettetem Metrikformat angeben, das generiert werden soll. Für jede Prometheus-Quelle, aus der der CloudWatch Agent standardmäßig importiert, gibt es `metric_declaration` Abschnitte. Diese Abschnitte enthalten jeweils die folgenden Felder:

- `label_matcher` ist ein regulärer Ausdruck, der den Wert der in `source_labels` aufgelisteten Beschriftungen überprüft. Die übereinstimmenden Metriken werden für die Aufnahme in das eingebettete Metrikformat aktiviert, an das gesendet wird. CloudWatch

  Wenn in `source_labels` mehrere Bezeichnungen angegeben sind, empfehlen wir, keine ^- oder $-Zeichen im regulären Ausdruck für `label_matcher` zu verwenden.
- `source_labels` gibt den Wert der Beschriftungen an, die von der `label_matcher`-Zeile überprüft werden.
- `label_separator` gibt das Trennzeichen an, das in der Zeile `label_matcher`verwendet werden soll, wenn mehrere `source_labels` angegeben werden. Der Standardwert ist `;`. Sie können diesen Standardwert in der Zeile `label_matcher` im folgenden Beispiel sehen.
- `metric_selectors`ist ein regulärer Ausdruck, der die Metriken angibt, die gesammelt und an sie gesendet werden sollen CloudWatch.
- `dimensions`ist die Liste der Bezeichnungen, die als CloudWatch Dimensionen für jede ausgewählte Metrik verwendet werden sollen.

Sehen Sie sich das folgende `metric_declaration`-Beispiel an.

```
"metric_declaration": [
  {
     "source_labels":[ "Service", "Namespace"],
     "label_matcher":"(.*node-exporter.*|.*kube-dns.*);kube-system",
     "dimensions":[
        ["Service", "Namespace"]
     ],
     "metric_selectors":[
        "^coredns_dns_request_type_count_total$"
     ]
  }
]
```

In diesem Beispiel wird ein eingebetteter Metrikformatabschnitt konfiguriert, der als Protokollereignis gesendet wird, wenn die folgenden Bedingungen erfüllt sind:

- Der Wert von `Service` enthält entweder `node-exporter` oder `kube-dns`.
- Der Wert von `Namespace` ist `kube-system`.

- Die Prometheus-Metrik `coredns_dns_request_type_count_total` enthält sowohl `Service`- als auch `Namespace`-Beschriftungen.

Das Protokollereignis, das gesendet wird, enthält den folgenden hervorgehobenen Abschnitt:

```
{
   "CloudWatchMetrics":[
      {
         "Metrics":[
            {
               "Name":"coredns_dns_request_type_count_total"
            }
         ],
         "Dimensions":[
            [
               "Namespace",
               "Service"
            ]
         ],
         "Namespace":"ContainerInsights/Prometheus"
      }
   ],
   "Namespace":"kube-system",
   "Service":"kube-dns",
   "coredns_dns_request_type_count_total":2562,
   "eks_amazonaws_com_component":"kube-dns",
   "instance":"192.168.61.254:9153",
   "job":"kubernetes-service-endpoints",
   ...
}
```

Tutorial zum Hinzufügen eines neuen Prometheus-Scrape-Ziels: Prometheus-API-Server-Metriken

Der Kubernetes API Server stellt Prometheus-Metriken standardmäßig auf Endpunkten zur Verfügung. Das offizielle Beispiel für die Kubernetes API Server-Scraping-Konfiguration ist auf Github verfügbar.

Das folgende Tutorial zeigt, wie Sie mit den folgenden Schritten beginnen, um mit dem Import von Kubernetes API-Server-Metriken zu beginnen: CloudWatch

- Hinzufügen der Prometheus-Scraping-Konfiguration für Kubernetes API Server zur Agenten-YAML-Datei. CloudWatch

- Konfiguration der Metrikdefinitionen im eingebetteten Metrikformat in der Agenten-YAML-Datei. CloudWatch
- (Optional) Erstellen eines CloudWatch Dashboards für die Kubernetes API-Server-Metriken.

> **Note**
>
> Der Kubernetes API Server stellt Mess-, Zähler-, Histogramm- und Übersichtsmetriken zur Verfügung. In dieser Version der Prometheus-Metrikunterstützung werden nur die Metriken mit den Typen Messgerät, Zähler und Zusammenfassung CloudWatch importiert.

Um mit der Erfassung von Kubernetes API Server Prometheus-Metriken zu beginnen in CloudWatch

1. Laden Sie die aktuelle Version der `prometheus-eks.yaml`-, `prometheus-eks-fargate.yaml`- oder `prometheus-k8s.yaml`-Datei herunter, indem Sie einen der folgenden Befehle eingeben.

   Geben Sie den folgenden Befehl für einen Amazon-EKS-Cluster mit dem EC2-Starttyp ein:

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks.yaml
   ```

   Geben Sie den folgenden Befehl für einen Amazon-EKS-Cluster mit dem Fargate-Starttyp ein:

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks-fargate.yaml
   ```

   Geben Sie für einen Kubernetes-Cluster, der auf einer Amazon-EC2-Instance ausgeführt wird, den folgenden Befehl ein:

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-k8s.yaml
   ```

2. Öffnen Sie die Datei mit einem Texteditor, suchen Sie den Abschnitt `prometheus-config` und fügen Sie den folgenden Abschnitt in diesem Abschnitt hinzu. Speichern Sie dann die Änderungen:

```
    # Scrape config for API servers
    - job_name: 'kubernetes-apiservers'
      kubernetes_sd_configs:
        - role: endpoints
          namespaces:
            names:
              - default
      scheme: https
      tls_config:
        ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
        insecure_skip_verify: true
      bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
      relabel_configs:
      - source_labels: [__meta_kubernetes_service_name,
__meta_kubernetes_endpoint_port_name]
        action: keep
        regex: kubernetes;https
      - action: replace
        source_labels:
        - __meta_kubernetes_namespace
        target_label: Namespace
      - action: replace
        source_labels:
        - __meta_kubernetes_service_name
        target_label: Service
```

3. Während Sie die YAML-Datei noch im Texteditor geöffnet haben, suchen Sie den Abschnitt `cwagentconfig.json`. Fügen Sie den folgenden Unterabschnitt hinzu und speichern Sie die Änderungen. In diesem Abschnitt werden die API-Server-Metriken in die Zulassungsliste für Agenten aufgenommen CloudWatch . Drei Typen von API-Server-Metriken werden der Genehmigungsliste hinzugefügt:

   - etcd-Objektanzahl
   - API-Server-Registrierungscontroller-Metriken
   - API-Server-Anforderungsmetriken

```
{"source_labels": ["job", "resource"],
  "label_matcher": "^kubernetes-apiservers;(services|daemonsets.apps|
deployments.apps|configmaps|endpoints|secrets|serviceaccounts|replicasets.apps)",
  "dimensions": [["ClusterName","Service","resource"]],
  "metric_selectors": [
  "^etcd_object_counts$"
  ]
},
{"source_labels": ["job", "name"],
   "label_matcher": "^kubernetes-apiservers;APIServiceRegistrationController$",
   "dimensions": [["ClusterName","Service","name"]],
   "metric_selectors": [
   "^workqueue_depth$",
   "^workqueue_adds_total$",
   "^workqueue_retries_total$"
  ]
},
{"source_labels": ["job","code"],
  "label_matcher": "^kubernetes-apiservers;2[0-9]{2}$",
  "dimensions": [["ClusterName","Service","code"]],
  "metric_selectors": [
   "^apiserver_request_total$"
  ]
},
{"source_labels": ["job"],
  "label_matcher": "^kubernetes-apiservers",
  "dimensions": [["ClusterName","Service"]],
  "metric_selectors": [
  "^apiserver_request_total$"
  ]
},
```

4. Wenn Sie den CloudWatch Agenten mit Prometheus-Unterstützung bereits im Cluster bereitgestellt haben, müssen Sie ihn löschen, indem Sie den folgenden Befehl eingeben:

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
```

5. Stellen Sie den CloudWatch Agenten mit Ihrer aktualisierten Konfiguration bereit, indem Sie einen der folgenden Befehle eingeben. Geben Sie für einen Amazon-EKS-Cluster mit dem Starttyp EC2 Folgendes ein:

```
kubectl apply -f prometheus-eks.yaml
```

Geben Sie den folgenden Befehl für einen Amazon-EKS-Cluster mit dem Fargate-Starttyp ein. Ersetzen Sie *MyCluster*und *Region* durch Werte, die Ihrer Bereitstellung entsprechen.

```
cat prometheus-eks-fargate.yaml \
| sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
| kubectl apply -f -
```

Geben Sie für einen Kubernetes-Cluster den folgenden Befehl ein. Ersetzen Sie *MyCluster*und *Region* durch Werte, die Ihrer Bereitstellung entsprechen.

```
cat prometheus-k8s.yaml \
| sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
| kubectl apply -f -
```

Danach sollten Sie einen neuen Protokoll-Stream namens kubernetes-apiservers in der Protokollgruppe /aws/containerinsights/***cluster_name***/prometheus sehen. Dieser Protokoll-Stream sollte Protokollereignisse mit einer Definition des eingebetteten Metrikformats wie folgt einschließen:

```
{
   "CloudWatchMetrics":[
      {
         "Metrics":[
            {
               "Name":"apiserver_request_total"
            }
         ],
         "Dimensions":[
            [
               "ClusterName",
               "Service"
            ]
         ],
         "Namespace":"ContainerInsights/Prometheus"
      }
   ],
   "ClusterName":"my-cluster-name",
   "Namespace":"default",
```

```
    "Service":"kubernetes",
    "Timestamp":"1592267020339",
    "Version":"0",
    "apiserver_request_count":0,
    "apiserver_request_total":0,
    "code":"0",
    "component":"apiserver",
    "contentType":"application/json",
    "instance":"192.0.2.0:443",
    "job":"kubernetes-apiservers",
    "prom_metric_type":"counter",
    "resource":"pods",
    "scope":"namespace",
    "verb":"WATCH",
    "version":"v1"
}
```

Sie können Ihre Metriken in der CloudWatch Konsole im ContainerInsights/Prometheus-Namespace anzeigen. Sie können optional auch ein CloudWatch Dashboard für Ihre Prometheus Kubernetes API Server-Metriken erstellen.

(Optional) Erstellen eines Dashboards für die Kubernetes API-Server-Metriken.

Um Kubernetes API Server-Metriken in Ihrem Dashboard zu sehen, müssen Sie zuerst die Schritte in den vorherigen Abschnitten ausgeführt haben, um mit der Erfassung dieser Metriken zu beginnen. CloudWatch

So erstellen Sie ein Dashboard für Kubernetes-API-Server-Metriken

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Vergewissern Sie sich, dass Sie die richtige AWS Region ausgewählt haben.
3. Wählen Sie im Navigationsbereich Dashboards aus.
4. Klicken Sie auf Create Dashboard (Dashboard erstellen). Geben Sie einen Namen für das neue Dashboard ein und wählen Sie Create dashboard (Dashboard erstellen).
5. Wählen Sie unter Add to this dashboard (Zu diesem Dashboard hinzufügen) die Option Cancel (Abbrechen).
6. Wählen Sie Actions (Aktionen), View/edit source (Quelle anzeigen/bearbeiten).
7. Laden Sie die folgende JSON-Datei herunter: Kubernetes API Dashboard-Quelle.

8. Öffnen Sie die heruntergeladene JSON-Datei mit einem Texteditor und nehmen Sie die folgenden Änderungen vor:

   - Ersetzen Sie alle `{{YOUR_CLUSTER_NAME}}`-Zeichenfolgen durch den genauen Namen Ihres Clusters. Stellen Sie sicher, dass keine Leerzeichen vor oder nach dem Text hinzugefügt werden.
   - Ersetzen Sie alle `{{YOUR_AWS_REGION}}`-Zeichenfolgen durch den Namen der Region, in der die Metriken erfasst werden. Zum Beispiel `us-west-2`. Stellen Sie sicher, dass keine Leerzeichen vor oder nach dem Text hinzugefügt werden.

9. Kopieren Sie den gesamten JSON-Blob und fügen Sie ihn in das Textfeld in der CloudWatch Konsole ein. Dabei wird der Inhalt des Felds ersetzt.
10. Wählen Sie Update (Aktualisieren), Save dashboard (Dashboard speichern).

(Optional) Einrichten von containerisierten Beispiel-Amazon-EKS-Workloads für Prometheus-Metrik-Tests

Um die Unterstützung von Prometheus-Metriken in CloudWatch Container Insights zu testen, können Sie einen oder mehrere der folgenden containerisierten Workloads einrichten. Der CloudWatch Agent mit Prometheus-Unterstützung sammelt automatisch Metriken von jeder dieser Workloads. Informationen zum Anzeigen der Metriken, die standardmäßig erfasst werden, finden Sie unter Vom Agenten gesammelte Prometheus-Metriken CloudWatch .

Bevor Sie einen dieser Workloads installieren können, müssen Sie Helm 3.x installieren, indem Sie die folgenden Befehle eingeben:

```
brew install helm
```

Weitere Informationen finden Sie unter Helm.

Themen

- AWS App Mesh -Beispiel-Workload für Amazon EKS und Kubernetes einrichten
- Einrichten von NGINX mit Beispielverkehr auf Amazon EKS und Kubernetes
- Einrichten von Memcached mit einem Metrik-Exporter auf Amazon EKS und Kubernetes
- Java/JMX-Beispiel-Workload für Amazon EKS und Kubernetes einrichten
- Einrichten von Memcached mit einem Metrik-Exporter auf Amazon EKS und Kubernetes

- Tutorial zum Hinzufügen eines neuen Prometheus-Scrape-Ziels: Redis auf Amazon-EKS- und Kubernetes-Clustern

AWS App Mesh -Beispiel-Workload für Amazon EKS und Kubernetes einrichten

Prometheus Prometheus-Unterstützung in CloudWatch Container Insights unterstützt. AWS App Mesh In den folgenden Abschnitten wird erläutert, wie Sie App Mesh einrichten.

CloudWatch Container Insights kann auch App Mesh Envoy Access Logs sammeln. Weitere Informationen finden Sie unter (Optional) App-Mesh-Envoy-Zugriffsprotokolle aktivieren.

Themen

- AWS App Mesh -Beispiel-Workload auf einem Amazon-EKS-Cluster mit dem Starttyp EC2 oder einem Kubernetes-Cluster einrichten
- Richten Sie einen AWS App Mesh Beispiel-Workload auf einem Amazon EKS-Cluster mit dem Starttyp Fargate ein

AWS App Mesh -Beispiel-Workload auf einem Amazon-EKS-Cluster mit dem Starttyp EC2 oder einem Kubernetes-Cluster einrichten

Gehen Sie wie folgt vor, wenn Sie App Mesh auf einem Cluster einrichten, auf dem Amazon EKS mit dem EC2-Starttyp, oder einem Kubernetes-Cluster ausgeführt wird.

Konfigurieren Sie IAM-Berechtigungen

Sie müssen die AWSAppMeshFullAccessRichtlinie zur IAM-Rolle für Ihre Amazon EKS- oder Kubernetes-Knotengruppe hinzufügen. Auf Amazon EKS sieht dieser Knotengruppenname ähnlich wie `eksctl-integ-test-eks-prometheus-NodeInstanceRole-ABCDEFHIJKL` aus. Auf Kubernetes könnte er ähnlich wie `nodes.integ-test-kops-prometheus.k8s.local` aussehen.

Installieren Sie App Mesh

Um den App-Mesh-Kubernetes-Controller zu installieren, befolgen Sie die Anweisungen in App-Mesh-Controller.

Installieren einer Beispielanwendung

aws-app-mesh-examplesenthält mehrere Komplettlösungen für Kubernetes App Mesh. In diesem Lernprogramm installieren Sie eine Beispielfarbanwendung, die zeigt, wie HTTP-Routen Header zum Abgleichen von eingehenden Anforderungen verwenden können.

So verwenden Sie eine App-Mesh-Beispielanwendung zum Testen von Container Insights

1. Installieren Sie die Anwendung mithilfe der folgenden Anweisungen: https://github.com/aws/aws-app-mesh-examples/tree/main/walkthroughs/howto-k8s-http-headers.
2. Starten Sie einen Curler-Pod, um Datenverkehr zu generieren:

   ```
   kubectl -n default run -it curler --image=tutum/curl /bin/bash
   ```

3. Curlen Sie verschiedene Endpunkte durch Ändern von HTTP-Headern. Führen Sie den Befehl curl mehrmals aus, wie gezeigt:

   ```
   curl -H "color_header: blue" front.howto-k8s-http-headers.svc.cluster.local:8080/;
    echo;

   curl -H "color_header: red" front.howto-k8s-http-headers.svc.cluster.local:8080/;
    echo;

   curl -H "color_header: yellow" front.howto-k8s-http-
   headers.svc.cluster.local:8080/; echo;
   ```

4. Öffnen Sie die Konsole unter https://console.aws.amazon.com/cloudwatch/ CloudWatch .
5. Wählen Sie in der AWS Region, in der Ihr Cluster ausgeführt wird, im Navigationsbereich die Option Metrics aus. Die Metrik befindet sich im ContainerInsights/Prometheus-Namespace.
6. Um die CloudWatch Logs-Ereignisse anzuzeigen, wählen Sie im Navigationsbereich Protokollgruppen aus. Die Ereignisse befinden sich in der Protokollgruppe `/aws/containerinsights/`*`your_cluster_name`*`/prometheus` im Protokollstream `kubernetes-pod-appmesh-envoy`.

Löschen der App-Mesh-Testumgebung

Wenn Sie mit der Verwendung von App Mesh und der Beispielanwendung fertig sind, verwenden Sie die folgenden Befehle, um die nicht benötigten Ressourcen zu löschen. Löschen Sie die Beispielanwendung, indem Sie den folgenden Befehl eingeben:

```
cd aws-app-mesh-examples/walkthroughs/howto-k8s-http-headers/
kubectl delete -f _output/manifest.yaml
```

Löschen Sie den App-Mesh-Controller, indem Sie den folgenden Befehl eingeben:

```
helm delete appmesh-controller -n appmesh-system
```

Richten Sie einen AWS App Mesh Beispiel-Workload auf einem Amazon EKS-Cluster mit dem Starttyp Fargate ein

Gehen Sie wie folgt vor, wenn Sie App Mesh auf einem Cluster einrichten, auf dem Amazon EKS mit dem Fargate-Starttyp ausgeführt wird.

Konfigurieren Sie IAM-Berechtigungen

Geben Sie den folgenden Befehl ein, um IAM-Berechtigungen einzurichten. *MyCluster*Ersetzen Sie es durch den Namen Ihres Clusters.

```
eksctl create iamserviceaccount --cluster MyCluster \
 --namespace howto-k8s-fargate \
 --name appmesh-pod \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshEnvoyAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapDiscoverInstanceAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSXRayDaemonWriteAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/CloudWatchLogsFullAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshFullAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapFullAccess \
 --override-existing-serviceaccounts \
 --approve
```

Installieren Sie App Mesh

Um den App-Mesh-Kubernetes-Controller zu installieren, befolgen Sie die Anweisungen in App-Mesh-Controller. Befolgen Sie unbedingt die Anweisungen für Amazon EKS mit dem Fargate-Starttyp.

Installieren einer Beispielanwendung

aws-app-mesh-examplesenthält mehrere Komplettlösungen für Kubernetes App Mesh. Für dieses Tutorial installieren Sie eine Beispielfarbanwendung, die für Amazon-EKS-Cluster mit dem Starttyp Fargate funktioniert.

So verwenden Sie eine App-Mesh-Beispielanwendung zum Testen von Container Insights

1. Installieren Sie die Anwendung mithilfe der folgenden Anweisungen: https://github.com/aws/aws-app-mesh-examples/tree/main/walkthroughs/howto-k8s-fargate.

   Bei diesen Anweisungen wird davon ausgegangen, dass Sie einen neuen Cluster mit dem korrekten Fargate-Profil erstellen. Wenn Sie einen Amazon-EKS-Cluster verwenden möchten, den Sie bereits eingerichtet haben, können Sie die folgenden Befehle verwenden, um diesen Cluster für diese Demonstration einzurichten. Ersetzen Sie es durch *MyCluster*den Namen Ihres Clusters.

   ```
   eksctl create iamserviceaccount --cluster MyCluster \
    --namespace howto-k8s-fargate \
    --name appmesh-pod \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshEnvoyAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapDiscoverInstanceAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSXRayDaemonWriteAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/CloudWatchLogsFullAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshFullAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapFullAccess \
    --override-existing-serviceaccounts \
    --approve
   ```

   ```
   eksctl create fargateprofile --cluster MyCluster \
   --namespace howto-k8s-fargate --name howto-k8s-fargate
   ```

2. Port-Weiterleitung der Front-Anwendungsbereitstellung:

   ```
   kubectl -n howto-k8s-fargate port-forward deployment/front 8080:8080
   ```

3. Curl der Front-App:

   ```
   while true; do  curl -s http://localhost:8080/color; sleep 0.1; echo ; done
   ```

4. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
5. Wählen Sie in der AWS Region, in der Ihr Cluster ausgeführt wird, im Navigationsbereich die Option Metrics aus. Die Metrik befindet sich im ContainerInsights/Prometheus-Namespace.
6. Um die CloudWatch Logs-Ereignisse anzuzeigen, wählen Sie im Navigationsbereich Protokollgruppen aus. Die Ereignisse befinden sich in der Protokollgruppe `/aws/`

`containerinsights/your_cluster_name/prometheus` im Protokollstream `kubernetes-pod-appmesh-envoy`.

Löschen der App-Mesh-Testumgebung

Wenn Sie mit der Verwendung von App Mesh und der Beispielanwendung fertig sind, verwenden Sie die folgenden Befehle, um die nicht benötigten Ressourcen zu löschen. Löschen Sie die Beispielanwendung, indem Sie den folgenden Befehl eingeben:

```
cd aws-app-mesh-examples/walkthroughs/howto-k8s-fargate/
kubectl delete -f _output/manifest.yaml
```

Löschen Sie den App-Mesh-Controller, indem Sie den folgenden Befehl eingeben:

```
helm delete appmesh-controller -n appmesh-system
```

Einrichten von NGINX mit Beispielverkehr auf Amazon EKS und Kubernetes

NGINX ist ein Webserver, der auch als Load Balancer und Reverse Proxy verwendet werden kann. Weitere Informationen darüber, wie Kubernetes NGINX für Ingress nutzt, finden Sie unter kubernetes/ingress-nginx.

So installieren Sie Ingress-NGINX mit einem Beispiels-Datenverkehrsservice zum Testen der Unterstützung für Container Insights Prometheus

1. Geben Sie den folgenden Befehl ein, um das Repo „Helm ingress-nginx“ hinzuzufügen.

   ```
   helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
   ```

2. Geben Sie die folgenden Befehle ein.

   ```
   kubectl create namespace nginx-ingress-sample

   helm install my-nginx ingress-nginx/ingress-nginx \
   --namespace nginx-ingress-sample \
   --set controller.metrics.enabled=true \
   --set-string controller.metrics.service.annotations."prometheus\.io/port"="10254" \
   --set-string controller.metrics.service.annotations."prometheus\.io/scrape"="true"
   ```

3. Überprüfen Sie, ob die Services korrekt gestartet wurden, indem Sie den folgenden Befehl eingeben:

```
kubectl get service -n nginx-ingress-sample
```

   Die Ausgabe dieses Befehls sollte mehrere Spalten, einschließlich einer `EXTERNAL-IP`-Spalte, anzeigen.

4. Setzen Sie eine `EXTERNAL-IP`-Variable auf den Wert der `EXTERNAL-IP`-Spalte in der Zeile des NGINX-Ingress-Controllers.

```
EXTERNAL_IP=your-nginx-controller-external-ip
```

5. Starten Sie NGINX-Beispieldatenverkehr, indem Sie den folgenden Befehl eingeben.

```
SAMPLE_TRAFFIC_NAMESPACE=nginx-sample-traffic
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_traffic/nginx-traffic/nginx-traffic-sample.yaml |
sed "s/{{external_ip}}/$EXTERNAL_IP/g" |
sed "s/{{namespace}}/$SAMPLE_TRAFFIC_NAMESPACE/g" |
kubectl apply -f -
```

6. Geben Sie den folgenden Befehl ein, um zu bestätigen, dass sich alle drei Pods im Status `Running` befinden.

```
kubectl get pod -n $SAMPLE_TRAFFIC_NAMESPACE
```

   Wenn sie ausgeführt werden, sollten Sie bald Metriken im ContainerInsights/Prometheus-Namespace sehen.

So deinstallieren Sie NGINX und die Beispieldatenverkehrsanwendung:

1. Löschen Sie den Beispielverkehrsservice, indem Sie den folgenden Befehl eingeben:

```
kubectl delete namespace $SAMPLE_TRAFFIC_NAMESPACE
```

2. Löschen Sie den NGINX-Ausgang nach dem Helm-Versionsnamen.

```
helm uninstall my-nginx --namespace nginx-ingress-sample
```

```
kubectl delete namespace nginx-ingress-sample
```

Einrichten von Memcached mit einem Metrik-Exporter auf Amazon EKS und Kubernetes

Memcached ist ein Open-Source-Speicherobjekt-Caching-System. Weitere Informationen finden Sie unter Was ist Memcached?.

Wenn Sie Memcached auf einem Cluster mit dem Starttyp Fargate ausführen, müssen Sie ein Fargate-Profil einrichten, bevor Sie die Schritte in diesem Verfahren ausführen. Geben Sie zum Einrichten des Profils den folgenden Befehl ein. Ersetzen Sie es durch *MyCluster*den Namen Ihres Clusters.

```
eksctl create fargateprofile --cluster MyCluster \
--namespace memcached-sample --name memcached-sample
```

So installieren Sie memcached mit einem Metrik-Exporter, um die Container Insights Prometheus-Unterstützung zu testen

1. Geben Sie den folgenden Befehl ein, um das Repo hinzuzufügen.

   ```
   helm repo add bitnami https://charts.bitnami.com/bitnami
   ```

2. Geben Sie den folgenden Befehl ein, um einen neuen Namespace zu erstellen.

   ```
   kubectl create namespace memcached-sample
   ```

3. Geben Sie den folgenden Befehl ein, um Memcached zu installieren.

   ```
   helm install my-memcached bitnami/memcached --namespace memcached-sample \
   --set metrics.enabled=true \
   --set-string serviceAnnotations.prometheus\\.io/port="9150" \
   --set-string serviceAnnotations.prometheus\\.io/scrape="true"
   ```

4. Geben Sie den folgenden Befehl ein, um die Anmerkung des ausgeführten Services zu prüfen:

   ```
   kubectl describe service my-memcached-metrics -n memcached-sample
   ```

   Sie sollten die folgenden zwei Anmerkungen sehen:

```
Annotations:   prometheus.io/port: 9150
               prometheus.io/scrape: true
```

So deinstallieren Sie memcached:

- Geben Sie die folgenden Befehle ein.

```
helm uninstall my-memcached --namespace memcached-sample
kubectl delete namespace memcached-sample
```

Java/JMX-Beispiel-Workload für Amazon EKS und Kubernetes einrichten

JMX Exporter ist ein offizieller Prometheus-Exporter, der JMX mBeans als Prometheus-Metriken erfassen und verfügbar machen kann. Weitere Informationen finden Sie unter prometheus/jmx_exporter.

Container Insights kann vordefinierte Prometheus-Metriken von Java Virtual Machine (JVM), Java und Tomcat (Catalina) mithilfe des JMX Exporter erfassen.

Standard-Prometheus-Scrape-Konfiguration

Standardmäßig scannt der CloudWatch Agent mit Prometheus-Unterstützung die Java/JMX-Prometheus-Metriken von jedem Pod in einem Amazon EKS- oder `http://CLUSTER_IP:9404/metrics` Kubernetes-Cluster. Dies geschieht durch `role: pod` Erkennung von Prometheus `kubernetes_sd_config`. 9404 ist der von Prometheus für JMX Exporter zugewiesene Standardport. Weitere Informationen zu `role: pod` Discovery finden Sie unter pod. Sie können den JMX Exporter so konfigurieren, dass die Metriken auf einem anderen Port oder metrics_path verfügbar gemacht werden. Wenn Sie den Port oder Pfad ändern, aktualisieren Sie die Standardeinstellung jmx scrape_config in der Agentenkonfigurationsübersicht. CloudWatch Führen Sie den folgenden Befehl aus, um die aktuelle Prometheus-Konfiguration des CloudWatch Agenten abzurufen:

```
kubectl describe cm prometheus-config -n amazon-cloudwatch
```

Die zu ändernden Felder sind die Felder `/metrics` und `regex: '.*:9404$'`, wie im folgenden Beispiel hervorgehoben.

```
job_name: 'kubernetes-jmx-pod'
sample_limit: 10000
metrics_path: /metrics
kubernetes_sd_configs:
- role: pod
relabel_configs:
- source_labels: [__address__]
  action: keep
  regex: '.*:9404$'
- action: replace
  regex: (.+)
  source_labels:
```

Andere Prometheus-Scrape-Konfiguration

Wenn Sie Ihre Anwendung, die auf einer Reihe von Pods mit Java/JMX-Prometheus-Exportern von einem Kubernetes-Service ausgeführt wird, verfügbar machen, können Sie auch zur Verwendung der `role: service`-Erkennung oder `role: endpoint`-Erkennung von Prometheus `kubernetes_sd_config` wechseln. Weitere Informationen zu diesen Ermittlungsmethoden finden Sie unter Service, Endpunkte und <kubernetes_sd_config>.

Diese beiden Service-Discovery-Modi bieten mehr Metalabels, die für Sie beim Erstellen der CloudWatch Metrik-Dimensionen nützlich sein könnten. Sie können beispielsweise `__meta_kubernetes_service_name` in `Service` umbenennen und in die Dimension Ihrer Metriken aufnehmen. Weitere Informationen zum Anpassen Ihrer CloudWatch Metriken und ihrer Dimensionen finden Sie unter. CloudWatch Agentenkonfiguration für Prometheus

Docker-Image mit JMX Exporter

Richten Sie anschließend ein Docker-Image ein. In den folgenden Abschnitten finden Sie zwei Beispiel-Dockerfiles.

Wenn Sie das Image erstellt haben, laden Sie es in Amazon EKS oder Kubernetes und führen Sie dann den folgenden Befehl aus, um zu überprüfen, ob Prometheus-Metriken von `JMX_EXPORTER` auf Port 9404 verfügbar gemacht werden. Ersetzen Sie *`$JAR_SAMPLE_TRAFFIC_POD`* durch den Namen des laufenden Pods und ersetzen Sie *`$JAR_SAMPLE_TRAFFIC_NAMESPACE`* durch Ihren Anwendungs-Namespace.

Wenn Sie JMX Exporter auf einem Cluster mit dem Fargate-Starttyp ausführen, müssen Sie auch ein Fargate-Profil einrichten, bevor Sie die Schritte in diesem Verfahren ausführen. Geben Sie zum

Einrichten des Profils den folgenden Befehl ein. Ersetzen Sie es *MyCluster*durch den Namen Ihres Clusters.

```
eksctl create fargateprofile --cluster MyCluster \
--namespace $JAR_SAMPLE_TRAFFIC_NAMESPACE\
 --name $JAR_SAMPLE_TRAFFIC_NAMESPACE
```

```
kubectl exec $JAR_SAMPLE_TRAFFIC_POD -n $JARCAT_SAMPLE_TRAFFIC_NAMESPACE -- curl
 http://localhost:9404
```

Beispiel: Apache-Tomcat-Docker-Image mit Prometheus-Metriken

Der Apache Tomcat-Server stellt JMX mBeans standardmäßig zur Verfügung. Sie können JMX Exporter mit Tomcat integrieren, um JMX mBeans als Prometheus-Metriken verfügbar zu machen. Das folgende Beispiel-Dockerfile zeigt die Schritte zum Erstellen eines Test-Images:

```
# From Tomcat 9.0 JDK8 OpenJDK
FROM tomcat:9.0-jdk8-openjdk

RUN mkdir -p /opt/jmx_exporter

COPY ./jmx_prometheus_javaagent-0.12.0.jar /opt/jmx_exporter
COPY ./config.yaml /opt/jmx_exporter
COPY ./setenv.sh /usr/local/tomcat/bin
COPY your web application.war /usr/local/tomcat/webapps/

RUN chmod  o+x /usr/local/tomcat/bin/setenv.sh

ENTRYPOINT ["catalina.sh", "run"]
```

In der folgenden Liste werden die vier COPY-Zeilen in diesem Dockerfile erläutert.

- Laden Sie die neueste JMX Exporter JAR-Datei von https://github.com/prometheus/jmx_exporter herunter.
- `config.yaml` ist die JMX Exporter-Konfigurationsdatei. Weitere Informationen finden Sie unter https://github.com/prometheus/jmx_exporter#Configuration.

  Hier ist eine Beispielkonfigurationsdatei für Java und Tomcat:

  ```
  lowercaseOutputName: true
  lowercaseOutputLabelNames: true
  ```

```
rules:
- pattern: 'java.lang<type=OperatingSystem><>(FreePhysicalMemorySize|
TotalPhysicalMemorySize|FreeSwapSpaceSize|TotalSwapSpaceSize|SystemCpuLoad|
ProcessCpuLoad|OpenFileDescriptorCount|AvailableProcessors)'
  name: java_lang_OperatingSystem_$1
  type: GAUGE

- pattern: 'java.lang<type=Threading><>(TotalStartedThreadCount|ThreadCount)'
  name: java_lang_threading_$1
  type: GAUGE

- pattern: 'Catalina<type=GlobalRequestProcessor, name=\"(\w+-\w+)-(\d+)\"><>(\w+)'
  name: catalina_globalrequestprocessor_$3_total
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina global $3
  type: COUNTER

- pattern: 'Catalina<j2eeType=Servlet, WebModule=//([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-
a-zA-Z0-9+&@#/%=~_|]), name=([-a-zA-Z0-9+/$%~_-|!.]*), J2EEApplication=none,
 J2EEServer=none><>(requestCount|maxTime|processingTime|errorCount)'
  name: catalina_servlet_$3_total
  labels:
    module: "$1"
    servlet: "$2"
  help: Catalina servlet $3 total
  type: COUNTER

- pattern: 'Catalina<type=ThreadPool, name="(\w+-\w+)-(\d+)"><>(currentThreadCount|
currentThreadsBusy|keepAliveCount|pollerThreadCount|connectionCount)'
  name: catalina_threadpool_$3
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina threadpool $3
  type: GAUGE

- pattern: 'Catalina<type=Manager, host=([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-a-zA-
Z0-9+&@#/%=~_|]), context=([-a-zA-Z0-9+/$%~_-|!.]*)><>(processingTime|sessionCounter|
rejectedSessions|expiredSessions)'
  name: catalina_session_$3_total
  labels:
```

```
    context: "$2"
    host: "$1"
  help: Catalina session $3 total
  type: COUNTER

- pattern: ".*"
```

- `setenv.sh` ist ein Tomcat-Startup-Skript zum Starten des JMX Exporter zusammen mit Tomcat, und um Prometheus-Metriken auf Port 9404 des lokalen Hosts verfügbar zu machen. Dazu übergibt es den `config.yaml`-Dateipfad an den JMX Exporter.

```
$ cat setenv.sh
export JAVA_OPTS="-javaagent:/opt/jmx_exporter/
jmx_prometheus_javaagent-0.12.0.jar=9404:/opt/jmx_exporter/config.yaml $JAVA_OPTS"
```

- Application.war ist Ihre Webanwendungs-`war`-Datei, die Tomcat zu finden hat.

Erstellen Sie ein Docker-Image mit dieser Konfiguration und laden Sie es in ein Image-Repository hoch.

Beispiel: Docker-Image der Java-Jar-Anwendung mit Prometheus-Metriken

Das folgende Beispiel-Dockerfile zeigt die Schritte zum Erstellen eines Test-Images:

```
# Alpine Linux with OpenJDK JRE
FROM openjdk:8-jre-alpine

RUN mkdir -p /opt/jmx_exporter

COPY ./jmx_prometheus_javaagent-0.12.0.jar /opt/jmx_exporter
COPY ./SampleJavaApplication-1.0-SNAPSHOT.jar /opt/jmx_exporter
COPY ./start_exporter_example.sh /opt/jmx_exporter
COPY ./config.yaml /opt/jmx_exporter

RUN chmod -R o+x /opt/jmx_exporter
RUN apk add curl

ENTRYPOINT exec /opt/jmx_exporter/start_exporter_example.sh
```

In der folgenden Liste werden die vier `COPY`-Zeilen in diesem Dockerfile erläutert.

- Laden Sie die neueste JMX Exporter JAR-Datei von https://github.com/prometheus/jmx_exporter herunter.
- `config.yaml` ist die JMX Exporter-Konfigurationsdatei. Weitere Informationen finden Sie unter https://github.com/prometheus/jmx_exporter#Configuration.

  Hier ist eine Beispielkonfigurationsdatei für Java und Tomcat:

```
lowercaseOutputName: true
lowercaseOutputLabelNames: true

rules:
- pattern: 'java.lang<type=OperatingSystem><>(FreePhysicalMemorySize|
TotalPhysicalMemorySize|FreeSwapSpaceSize|TotalSwapSpaceSize|SystemCpuLoad|
ProcessCpuLoad|OpenFileDescriptorCount|AvailableProcessors)'
  name: java_lang_OperatingSystem_$1
  type: GAUGE

- pattern: 'java.lang<type=Threading><>(TotalStartedThreadCount|ThreadCount)'
  name: java_lang_threading_$1
  type: GAUGE

- pattern: 'Catalina<type=GlobalRequestProcessor, name=\"(\w+-\w+)-(\d+)\"><>(\w+)'
  name: catalina_globalrequestprocessor_$3_total
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina global $3
  type: COUNTER

- pattern: 'Catalina<j2eeType=Servlet, WebModule=//([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-
a-zA-Z0-9+&@#/%=~_|]), name=([-a-zA-Z0-9+/$%~_-|!.]*), J2EEApplication=none,
 J2EEServer=none><>(requestCount|maxTime|processingTime|errorCount)'
  name: catalina_servlet_$3_total
  labels:
    module: "$1"
    servlet: "$2"
  help: Catalina servlet $3 total
  type: COUNTER

- pattern: 'Catalina<type=ThreadPool, name="(\w+-\w+)-(\d+)"><>(currentThreadCount|
currentThreadsBusy|keepAliveCount|pollerThreadCount|connectionCount)'
  name: catalina_threadpool_$3
  labels:
```

```
    port: "$2"
    protocol: "$1"
  help: Catalina threadpool $3
  type: GAUGE

- pattern: 'Catalina<type=Manager, host=([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-a-zA-
Z0-9+&@#/%=~_|]), context=([-a-zA-Z0-9+/$%~_-|!.]*)><>(processingTime|sessionCounter|
rejectedSessions|expiredSessions)'
  name: catalina_session_$3_total
  labels:
    context: "$2"
    host: "$1"
  help: Catalina session $3 total
  type: COUNTER

- pattern: ".*"
```

- `start_exporter_example.sh` ist das Skript zum Starten der JAR-Anwendung mit den exportierten Prometheus-Metriken. Dazu übergibt es den `config.yaml`-Dateipfad an den JMX Exporter.

```
$ cat start_exporter_example.sh
java -javaagent:/opt/jmx_exporter/jmx_prometheus_javaagent-0.12.0.jar=9404:/
opt/jmx_exporter/config.yaml -cp  /opt/jmx_exporter/SampleJavaApplication-1.0-
SNAPSHOT.jar com.gubupt.sample.app.App
```

- SampleJavaApplication-1.0-Snapshot.jar ist die JAR-Beispieldatei für eine Java-Anwendung. Ersetzen Sie sie durch die Java-Anwendung, die Sie überwachen möchten.

Erstellen Sie ein Docker-Image mit dieser Konfiguration und laden Sie es in ein Image-Repository hoch.

Einrichten von Memcached mit einem Metrik-Exporter auf Amazon EKS und Kubernetes

HAProxy ist eine Open-Source-Proxy-Anwendung. Weitere Informationen finden Sie unter HAProxy.

Wenn Sie HAProxy auf einem Cluster mit dem Starttyp Fargate ausführen, müssen Sie ein Fargate-Profil einrichten, bevor Sie die Schritte in diesem Verfahren ausführen. Geben Sie zum Einrichten des Profils den folgenden Befehl ein. Ersetzen Sie durch den Namen Ihres *MyCluster*Clusters.

```
eksctl create fargateprofile --cluster MyCluster \
```

```
--namespace haproxy-ingress-sample --name haproxy-ingress-sample
```

So installieren Sie HAProxy mit einem Metrik-Exporter, um die Container Insights Prometheus-Unterstützung zu testen:

1. Geben Sie den folgenden Befehl ein, um den Helm-Incubator-Repo hinzuzufügen:

   ```
   helm repo add haproxy-ingress https://haproxy-ingress.github.io/charts
   ```

2. Geben Sie den folgenden Befehl ein, um einen neuen Namespace zu erstellen.

   ```
   kubectl create namespace haproxy-ingress-sample
   ```

3. Geben Sie die folgenden Befehle ein, um HAProxy zu installieren:

   ```
   helm install haproxy haproxy-ingress/haproxy-ingress \
   --namespace haproxy-ingress-sample \
   --set defaultBackend.enabled=true \
   --set controller.stats.enabled=true \
   --set controller.metrics.enabled=true \
   --set-string controller.metrics.service.annotations."prometheus\.io/port"="9101" \
   --set-string controller.metrics.service.annotations."prometheus\.io/scrape"="true"
   ```

4. Geben Sie den folgenden Befehl ein, um die Anmerkung des Services zu bestätigen:

   ```
   kubectl describe service haproxy-haproxy-ingress-metrics -n haproxy-ingress-sample
   ```

   Sie sollten die folgenden Anmerkungen sehen.

   ```
   Annotations:   prometheus.io/port: 9101
                  prometheus.io/scrape: true
   ```

So deinstallieren Sie HAProxy:

- Geben Sie die folgenden Befehle ein.

  ```
  helm uninstall haproxy --namespace haproxy-ingress-sample
  kubectl delete namespace haproxy-ingress-sample
  ```

Tutorial zum Hinzufügen eines neuen Prometheus-Scrape-Ziels: Redis auf Amazon-EKS- und Kubernetes-Clustern

Dieses Tutorial bietet eine praktische Einführung zum Scraping der Prometheus-Metriken einer Redis-Beispielanwendung auf Amazon EKS und Kubernetes. Redis (https://redis.io/) ist ein Open Source (BSD lizenziert), In-Memory-Datenstrukturspeicher, der als Datenbank, Cache und Message Broker verwendet wird. Weitere Informationen finden Sie unter redis.

redis_exporter (mit MIT-Lizenz lizenziert) wird verwendet, um die Redis-Prometheus-Metriken auf dem angegebenen Port verfügbar zu machen (Standard: 0.0.0.0:9121). Weitere Informationen finden Sie unter redis_exporter.

Die Docker-Images in den folgenden zwei Docker Hub-Repositories werden in diesem Tutorial verwendet:

- redis
- redis_exporter

So installieren Sie eine Redis-Beispiel-Workload, die Prometheus-Metriken verfügbar macht

1. Legen Sie den Namespace für die Redis-Beispiel-Workload fest.

   ```
   REDIS_NAMESPACE=redis-sample
   ```

2. Wenn Sie Redis auf einem Cluster mit dem Starttyp Fargate ausführen, müssen Sie ein Fargate-Profil einrichten. Geben Sie zum Einrichten des Profils den folgenden Befehl ein. *MyCluster*Ersetzen Sie durch den Namen Ihres Clusters.

   ```
   eksctl create fargateprofile --cluster MyCluster \
   --namespace $REDIS_NAMESPACE --name $REDIS_NAMESPACE
   ```

3. Geben Sie den folgenden Befehl ein, um die Beispiel-Workload zu installieren.

   ```
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
   insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
   prometheus/sample_traffic/redis/redis-traffic-sample.yaml \
   | sed "s/{{namespace}}/$REDIS_NAMESPACE/g" \
   | kubectl apply -f -
   ```

4. Die Installation umfasst einen Service namens `my-redis-metrics`, der die Redis-Prometheus-Metrik auf Port 9121 verfügbar macht. Geben Sie den folgenden Befehl ein, um die Details des Services abzurufen:

   ```
   kubectl describe service/my-redis-metrics  -n $REDIS_NAMESPACE
   ```

   Im `Annotations` Bereich der Ergebnisse sehen Sie zwei Anmerkungen, die der Prometheus-Scrape-Konfiguration des CloudWatch Agenten entsprechen, sodass er die Workloads automatisch erkennen kann:

   ```
   prometheus.io/port: 9121
   prometheus.io/scrape: true
   ```

   Die zugehörige Prometheus-Scrape-Konfiguration finden Sie im `- job_name: kubernetes-service-endpoints`-Abschnitt von `kubernetes-eks.yaml` oder `kubernetes-k8s.yaml`.

Um mit der Erfassung von Redis Prometheus-Metriken zu beginnen in CloudWatch

1. Laden Sie die aktuelle Version der `kubernetes-eks.yaml`- oder `kubernetes-k8s.yaml`-Datei herunter, indem Sie einen der folgenden Befehle eingeben. Geben Sie diesen Befehl für einen Amazon-EKS-Cluster mit dem EC2-Starttyp ein.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks.yaml
   ```

   Geben Sie diesen Befehl für einen Amazon-EKS-Cluster mit dem Fargate-Starttyp ein.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks-fargate.yaml
   ```

   Geben Sie für einen Kubernetes-Cluster, der auf einer Amazon-EC2-Instance ausgeführt wird, diesen Befehl ein.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-k8s.yaml
```

2. Öffnen Sie die Datei mit einem Texteditor und suchen Sie den Abschnitt `cwagentconfig.json`. Fügen Sie den folgenden Unterabschnitt hinzu und speichern Sie die Änderungen. Stellen Sie sicher, dass die Einrückung dem vorhandenen Muster folgt.

```
{
  "source_labels": ["pod_name"],
  "label_matcher": "^redis-instance$",
  "dimensions": [["Namespace","ClusterName"]],
  "metric_selectors": [
    "^redis_net_(in|out)put_bytes_total$",
    "^redis_(expired|evicted)_keys_total$",
    "^redis_keyspace_(hits|misses)_total$",
    "^redis_memory_used_bytes$",
    "^redis_connected_clients$"
  ]
},
{
  "source_labels": ["pod_name"],
  "label_matcher": "^redis-instance$",
  "dimensions": [["Namespace","ClusterName","cmd"]],
  "metric_selectors": [
    "^redis_commands_total$"
  ]
},
{
  "source_labels": ["pod_name"],
  "label_matcher": "^redis-instance$",
  "dimensions": [["Namespace","ClusterName","db"]],
  "metric_selectors": [
    "^redis_db_keys$"
  ]
},
```

In dem Abschnitt, den Sie hinzugefügt haben, werden die Redis-Metriken in die CloudWatch Zulassungsliste für Agenten aufgenommen. Eine Liste dieser Metriken finden Sie im folgenden Abschnitt.

3. Wenn Sie den CloudWatch Agenten mit Prometheus-Unterstützung bereits in diesem Cluster bereitgestellt haben, müssen Sie ihn löschen, indem Sie den folgenden Befehl eingeben.

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
```

4. Stellen Sie den CloudWatch Agenten mit Ihrer aktualisierten Konfiguration bereit, indem Sie einen der folgenden Befehle eingeben. Ersetzen Sie *MyCluster*und *wählen Sie die Region* entsprechend Ihren Einstellungen aus.

   Geben Sie diesen Befehl für einen Amazon-EKS-Cluster mit dem EC2-Starttyp ein.

```
kubectl apply -f prometheus-eks.yaml
```

   Geben Sie diesen Befehl für einen Amazon-EKS-Cluster mit dem Fargate-Starttyp ein.

```
cat prometheus-eks-fargate.yaml \
| sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
| kubectl apply -f -
```

   Geben Sie für einen Kubernetes-Cluster folgenden Befehl ein.

```
cat prometheus-k8s.yaml \
| sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
| kubectl apply -f -
```

Anzeigen Ihrer Redis-Prometheus-Metriken

Dieses Tutorial sendet die folgenden Metriken an den ContainerInsights/Prometheus-Namespace in. CloudWatch Sie können die CloudWatch Konsole verwenden, um die Metriken in diesem Namespace zu sehen.

Metrikname	Dimensionen	
`redis_net_input_bytes_total`	ClusterName, `Namespace`	

Metrikname	Dimensionen	
redis_net_output_bytes_total	ClusterName, Namespace	
redis_expired_keys_total	ClusterName, Namespace	
redis_evicted_keys_total	ClusterName, Namespace	
redis_keyspace_hits_total	ClusterName, Namespace	
redis_keyspace_misses_total	ClusterName, Namespace	
redis_memory_used_bytes	ClusterName, Namespace	
redis_connected_clients	ClusterName, Namespace	
redis_commands_total	ClusterName,, Namespace cmd	
redis_db_keys	ClusterName,Namespace , db	

> **Note**
>
> Die Werte der cmd-Dimension können append, client, command, config, dbsize, flushall, get, incr, info, latency oder slowlog sein.

Die Werte der Db-Dimension können `db0` oder `db15` sein.

Sie können auch ein CloudWatch Dashboard für Ihre Redis Prometheus-Metriken erstellen.

So erstellen Sie ein Dashboard für Redis-Prometheus-Metriken

1. Erstellen Sie Umgebungsvariablen und ersetzen Sie die folgenden Werte entsprechend Ihrer Bereitstellung.

```
DASHBOARD_NAME=your_cw_dashboard_name
REGION_NAME=your_metric_region_such_as_us-east-1
CLUSTER_NAME=your_k8s_cluster_name_here
NAMESPACE=your_redis_service_namespace_here
```

2. Verwenden Sie den folgenden Befehl, um das Dashboard zu erstellen.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_cloudwatch_dashboards/redis/cw_dashboard_redis.json \
| sed "s/{{YOUR_AWS_REGION}}/${REGION_NAME}/g" \
| sed "s/{{YOUR_CLUSTER_NAME}}/${CLUSTER_NAME}/g" \
| sed "s/{{YOUR_NAMESPACE}}/${NAMESPACE}/g" \
```

Konvertierung des metrischen Prometheus-Typs durch den Agenten CloudWatch

Die Prometheus-Clientbibliotheken bieten vier Kernmetriktypen:

- Zähler
- Messinstrument
- Übersicht
- Histogramm

Der CloudWatch Agent unterstützt die Metriktypen Zähler, Messgerät und Zusammenfassung. Unterstützung für Histogrammmetriken ist für eine kommende Version geplant.

Die Prometheus-Metriken mit dem Metriktyp Histogramm werden vom Agenten gelöscht. CloudWatch Weitere Informationen finden Sie unter Protokollierung von gelöschten Prometheus-Metriken.

Messinstrumentmetriken

Eine Prometheus-Messwertmetrik ist eine Metrik, die einen einzelnen numerischen Wert darstellt, der beliebig nach oben und unten gehen kann. Der CloudWatch Agent erfasst Messmetriken und sendet diese Werte direkt aus.

Zähler-Metriken

Eine Prometheus-Zählermetrik ist eine kumulative Metrik, die einen einzelnen monotonisch zunehmenden Zähler darstellt, dessen Wert nur erhöht oder auf Null zurückgesetzt werden kann. Der CloudWatch Agent berechnet ein Delta aus dem vorherigen Scrape und sendet den Deltawert als Metrikwert in das Protokollereignis. Der CloudWatch Agent beginnt also, ab dem zweiten Scrape ein Log-Ereignis zu erzeugen und setzt, falls vorhanden, mit nachfolgenden Scrapes fort.

Zusammenfassende Metriken

Eine Prometheus-Zusammenfassungsmetrik ist ein komplexer Metriktyp, der durch mehrere Datenpunkte dargestellt wird. Es liefert eine Gesamtzahl der Beobachtungen und eine Summe aller beobachteten Werte. Er berechnet konfigurierbare Quantile über ein Schiebezeitfenster.

Die Summe und Anzahl einer Zusammenfassungsmetrik sind kumulativ, die Quantile jedoch nicht. Das folgende Beispiel veranschaulicht die Varianz von Quantilen.

```
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 7.123e-06
go_gc_duration_seconds{quantile="0.25"} 9.204e-06
go_gc_duration_seconds{quantile="0.5"} 1.1065e-05
go_gc_duration_seconds{quantile="0.75"} 2.8731e-05
go_gc_duration_seconds{quantile="1"} 0.003841496
go_gc_duration_seconds_sum 0.37630427
go_gc_duration_seconds_count 9774
```

Der CloudWatch Agent verarbeitet die Summe und Anzahl einer Übersichtsmetrik genauso wie Zählermetriken, wie im vorherigen Abschnitt beschrieben. Der CloudWatch Agent behält die Quantilwerte so bei, wie sie ursprünglich gemeldet wurden.

Vom Agenten gesammelte Prometheus-Metriken CloudWatch

Der CloudWatch Agent mit Prometheus-Unterstützung sammelt automatisch Metriken von verschiedenen Diensten und Workloads. Die Metriken, die standardmäßig erfasst werden, sind in den folgenden Abschnitten aufgeführt. Sie können den Agenten auch so konfigurieren, dass er

weitere Metriken von diesen Services erfasst und Prometheus-Metriken von anderen Anwendungen und Services erfasst. Weitere Informationen zum Erfassen zusätzlicher Metriken finden Sie unter CloudWatch Agentenkonfiguration für Prometheus.

Prometheus-Metriken, die von Amazon EKS- und Kubernetes-Clustern gesammelt wurden, befinden sich im /Prometheus-Namespace. ContainerInsights Aus Amazon ECS-Clustern gesammelte Prometheus-Metriken befinden sich im ContainerInsightsECS/Prometheus-Namespace.

Themen

Prometheus-Metriken für App Mesh

Die folgenden Metriken werden automatisch von App Mesh erfasst.

CloudWatch Container Insights kann auch App Mesh Envoy Access Logs sammeln. Weitere Informationen finden Sie unter (Optional) App-Mesh-Envoy-Zugriffsprotokolle aktivieren.

Prometheus-Metriken für App Mesh in Amazon-EKS- und Kubernetes-Clustern

Metrikname	Dimensionen	
`envoy_http_downstream_rq_total`	ClusterName, `Namespace`	
`envoy_http_downstream_rq_xx`	ClusterName, `Namespace` ClusterName,, `Namespace` envoy_http_conn_manager_prefix, envoy_response_code_class	
`envoy_cluster_upst`	ClusterName, `Namespace`	

Metrikname	Dimensionen	
ream_cx_r x_bytes_total		
envoy_clu ster_upst ream_cx_t x_bytes_total	ClusterName, Namespace	
envoy_clu ster_memb ership_healthy	ClusterName, Namespace	
envoy_clu ster_memb ership_total	ClusterName, Namespace	
envoy_ser ver_memor y_heap_size	ClusterName, Namespace	
envoy_ser ver_memor y_allocated	ClusterName, Namespace	
envoy_clu ster_upst ream_cx_c onnect_timeout	ClusterName, Namespace	
envoy_clu ster_upst ream_rq_p ending_fa ilure_eject	ClusterName, Namespace	

Metrikname	Dimensionen	
envoy_cluster_upstream_rq_pending_overflow	ClusterName, Namespace	
envoy_cluster_upstream_rq_timeout	ClusterName, Namespace	
envoy_cluster_upstream_rq_try_per_timeout	ClusterName, Namespace	
envoy_cluster_upstream_rq_rx_reset	ClusterName, Namespace	
envoy_cluster_upstream_cx_destroy_local_with_active_rq	ClusterName, Namespace	
envoy_cluster_upstream_cx_destroy_remote_active_rq	ClusterName, Namespace	

Metrikname	Dimensionen	
envoy_cluster_upstream_rq_maintenance_mode	ClusterName, Namespace	
envoy_cluster_upstream_flow_control_paused_reading_total	ClusterName, Namespace	
envoy_cluster_upstream_flow_control_resumed_reading_total	ClusterName, Namespace	
envoy_cluster_upstream_flow_control_backed_up_total	ClusterName, Namespace	
envoy_cluster_upstream_flow_control_drained_total	ClusterName, Namespace	

Metrikname	Dimensionen	
envoy_cluster_upstream_rq_retry	ClusterName, Namespace	
envoy_cluster_upstream_rq_retry_success	ClusterName, Namespace	
envoy_cluster_upstream_rq_retry_overflow	ClusterName, Namespace	
envoy_server_live	ClusterName, Namespace	
envoy_server_uptime	ClusterName, Namespace	

Prometheus-Metriken für App Mesh auf Amazon-ECS-Clustern

Metrikname	Dimensionen	
envoy_http_downstream_rq_total	ClusterName, TaskDefinitionFamily	
envoy_http_downstream_rq_xx	ClusterName, TaskDefinitionFamily	
envoy_cluster_upst	ClusterName, TaskDefinitionFamily	

Metrikname	Dimensionen	
ream_cx_r x_bytes_total		
envoy_clu ster_upst ream_cx_t x_bytes_total	ClusterName, TaskDefinitionFamily	
envoy_clu ster_memb ership_healthy	ClusterName, TaskDefinitionFamily	
envoy_clu ster_memb ership_total	ClusterName, TaskDefinitionFamily	
envoy_ser ver_memor y_heap_size	ClusterName, TaskDefinitionFamily	
envoy_ser ver_memor y_allocated	ClusterName, TaskDefinitionFamily	
envoy_clu ster_upst ream_cx_c onnect_timeout	ClusterName, TaskDefinitionFamily	
envoy_clu ster_upst ream_rq_p ending_fa ilure_eject	ClusterName, TaskDefinitionFamily	

Metrikname	Dimensionen	
envoy_cluster_upstream_rq_pending_overflow	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_timeout	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_try_per_timeout	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_rx_reset	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_cx_destroy_local_with_active_rq	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_cx_destroy_remote_active_rq	ClusterName, TaskDefinitionFamily	

Metrikname	Dimensionen	
envoy_cluster_upstream_rq_maintenance_mode	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_flow_control_paused_reading_total	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_flow_control_resumed_reading_total	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_flow_control_backed_up_total	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_flow_control_drained_total	ClusterName, TaskDefinitionFamily	

Metrikname	Dimensionen
envoy_cluster_upstream_rq_retry	ClusterName, TaskDefinitionFamily
envoy_cluster_upstream_rq_retry_success	ClusterName, TaskDefinitionFamily
envoy_cluster_upstream_rq_retry_overflow	ClusterName, TaskDefinitionFamily
envoy_server_live	ClusterName, TaskDefinitionFamily
envoy_server_uptime	ClusterName, TaskDefinitionFamily
envoy_http_downstream_rq_xx	ClusterName TaskDefinitionFamily, envoy_http_conn_manager_prefix, envoy_response_code_class ClusterName, TaskDefinitionFamily, envoy_response_code_class

Note

`TaskDefinitionFamily` ist der Kubernetes-Namespace des Netzes.
Folgende Werte von `envoy_http_conn_manager_prefix` sind möglich: `ingress`, `egress` oder `admin`.
Der Wert von `envoy_response_code_class` kann 1 (steht für `1xx`), 2 steht für `2xx`), 3steht für `3xx`), 4 steht für `4xx`), oder 5 steht für `5xx`) sein.

Prometheus-Metriken für NGINX

Die folgenden Metriken werden automatisch von NGINX auf Amazon-EKS- und Kubernetes-Clustern erfasst.

Metrikname	Dimensionen	
`nginx_ingress_controller_nginx_process_cpu_seconds_total`	ClusterNameNamespace ,, Dienst	
`nginx_ingress_controller_success`	ClusterNameNamespace , Bedienung	
`nginx_ingress_controller_requests`	ClusterNameNamespace , Bedienung	
`nginx_ingress_controller_nginx_process_connections`	ClusterNameNamespace , Bedienung	
`nginx_ingress_controller_nginx_process_connections_total`	ClusterNameNamespace , Bedienung	
`nginx_ingress_cont`	ClusterNameNamespace , Bedienung	

Metrikname	Dimensionen	
roller_nginx_process_resident_memory_bytes		
nginx_ingress_controller_config_last_reload_successful	ClusterNameNamespace , Bedienung	
nginx_ingress_controller_requests	ClusterNameNamespace , Dienst, Status	

Prometheus-Metriken für Memcached

Die folgenden Metriken werden automatisch von Memcached auf Amazon-EKS- und Kubernetes-Clustern erfasst.

Metrikname	Dimensionen	
memcached_current_items	ClusterNameNamespace , Dienst	
memcached_current_connections	ClusterNameNamespace , Bedienung	
memcached_limit_bytes	ClusterNameNamespace , Bedienung	

Metrikname	Dimensionen	
memcached_current_bytes	ClusterNameNamespace , Bedienung	
memcached_written_bytes_total	ClusterNameNamespace , Bedienung	
memcached_read_bytes_total	ClusterNameNamespace , Bedienung	
memcached_items_evicted_total	ClusterNameNamespace , Bedienung	
memcached_items_reclaimed_total	ClusterNameNamespace , Bedienung	
memcached_commands_total	ClusterNameNamespace , Bedienung ClusterNameNamespace , Service, Befehl ClusterNameNamespace , Dienst, Status, Befehl	

Prometheus-Metriken für Java/JMX

Metriken, die auf Amazon-EKS- und Kubernetes-Clustern erfasst werden

Auf Amazon-EKS- und Kubernetes-Clustern kann Container Insights mit dem JMX-Exporter die folgenden vordefinierten Prometheus-Metriken von Java Virtual Machine (JVM), Java und Tomcat (Catalina) sammeln. Weitere Informationen finden Sie unter prometheus/jmx_exporter auf Github.

Java/JMX auf Amazon-EKS- und Kubernetes-Clustern

Metrikname	Dimensionen
jvm_classes_loaded	ClusterName , Namespace
jvm_threads_current	ClusterName , Namespace
jvm_threads_daemon	ClusterName , Namespace
java_lang_operatingsystem_totalswapspacesize	ClusterName , Namespace
java_lang_operatingsystem_systemcpuload	ClusterName , Namespace
java_lang_operatingsystem_processcpuload	ClusterName , Namespace
java_lang_operatingsystem_freeswapspacesize	ClusterName , Namespace
java_lang_operatingsystem_totalphysicalmemorysize	ClusterName , Namespace

Metrikname	Dimensionen	
java_lang_operatingsystem_freephysicalmemorysize	ClusterName , Namespace	
java_lang_operatingsystem_openfiledescriptorcount	ClusterName , Namespace	
java_lang_operatingsystem_availableprocessors	ClusterName , Namespace	
jvm_memory_bytes_used	ClusterName , Namespace , Bereich	
jvm_memory_pool_bytes_used	ClusterName , Namespace , Pool	

> **Note**
>
> Die Werte der `area`-Dimension können `heap` oder sein `nonheap`.
> Die Werte der `pool`-Dimension können `Tenured Gen`, `Compress Class Space`, `Survivor Space`, `Eden Space`, `Code Cache` oder `Metaspace` sein.

Tomcat/JMX auf Amazon-EKS- und Kubernetes-Clustern

Zusätzlich zu den Java/JMX-Metriken in der vorherigen Tabelle werden auch die folgenden Metriken für den Tomcat-Workload erfasst.

Metrikname	Dimensionen	
catalina_manager_activesessions	ClusterName , Namespace	
catalina_manager_rejectedsessions	ClusterName , Namespace	
catalina_globalrequestprocessor_bytesreceived	ClusterName , Namespace	
catalina_globalrequestprocessor_bytessent	ClusterName , Namespace	
catalina_globalrequestprocessor_requestcount	ClusterName , Namespace	
catalina_globalrequestprocessor_errorcount	ClusterName , Namespace	

Metrikname	Dimensionen	
catalina_globalrequestprocessor_processingtime	ClusterName , Namespace	

Java/JMX auf Amazon-ECS-Clustern

Metrikname	Dimensionen	
jvm_classes_loaded	ClusterName , TaskDefinitionFamily	
jvm_threads_current	ClusterName , TaskDefinitionFamily	
jvm_threads_daemon	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_totalswapspacesize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_systemcpuload	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_processcpuload	ClusterName , TaskDefinitionFamily	

Metrikname	Dimensionen	
java_lang_operatingsystem_freeswapspacesize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_totalphysicalmemorysize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_freephysicalmemorysize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_openfiledescriptorcount	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_availableprocessors	ClusterName , TaskDefinitionFamily	
jvm_memory_bytes_used	ClusterName TaskDefinitionFamily, Bereich	
jvm_memory_pool_bytes_used	ClusterName TaskDefinitionFamily, Schwimmbad	

> **Note**
>
> Die Werte der `area`-Dimension können `heap` oder sein `nonheap`.
> Die Werte der `pool`-Dimension können `Tenured Gen`, `Compress Class Space`, `Survivor Space`, `Eden Space`, `Code Cache` oder `Metaspace` sein.

Tomcat/JMX auf Amazon-ECS-Clustern

Zusätzlich zu den Java/JMX-Metriken in der vorherigen Tabelle werden auch die folgenden Metriken für die Tomcat-Workload auf Amazon-ECS-Clustern erfasst.

Metrikname	Dimensionen	
`catalina_manager_activesessions`	`ClusterName` , `TaskDefinitionFamily`	
`catalina_manager_rejectedsessions`	`ClusterName` , `TaskDefinitionFamily`	
`catalina_globalrequestprocessor_bytesreceived`	`ClusterName` , `TaskDefinitionFamily`	
`catalina_globalrequestprocessor_bytessent`	`ClusterName` , `TaskDefinitionFamily`	
`catalina_globalrequestproce`	`ClusterName` , `TaskDefinitionFamily`	

Metrikname	Dimensionen	
ssor_requestcount		
catalina_globalrequestprocessor_errorcount	ClusterName , TaskDefinitionFamily	
catalina_globalrequestprocessor_processingtime	ClusterName , TaskDefinitionFamily	

Prometheus-Metriken für HAProxy

Die folgenden Metriken werden automatisch von HAProxy auf Amazon-EKS- und Kubernetes-Clustern erfasst.

Die erfassten Metriken hängen davon ab, welche Version von HAProxy Ingress Sie verwenden. Weitere Informationen zu HAProxy Ingress und seinen Versionen finden Sie unter haproxy-ingress.

Metrikname	Dimensionen	Verfügbarkeit
haproxy_backend_bytes_in_total	ClusterName , Namespace , Service	Alle Versionen von HAProxy Ingress
haproxy_backend_bytes_out_total	ClusterName , Namespace , Service	Alle Versionen von HAProxy Ingress
haproxy_backend_co	ClusterName , Namespace , Service	Alle Versionen von HAProxy Ingress

Metrikname	Dimensionen	Verfügbarkeit
nnection_ errors_total		
haproxy_b ackend_co nnections _total	ClusterName , Namespace , Service	Alle Versionen von HAProxy Ingress
haproxy_b ackend_cu rrent_sessions	ClusterName , Namespace , Service	Alle Versionen von HAProxy Ingress
haproxy_b ackend_ht tp_respon ses_total	ClusterName , Namespace , Service, Code, Backend	Alle Versionen von HAProxy Ingress
haproxy_b ackend_status	ClusterName , Namespace , Service	Nur in Versionen 0.10 oder höher von HAProxy Ingress
haproxy_b ackend_up	ClusterName , Namespace , Service	Nur in Versionen von HAProxy Ingress vor 0.10
haproxy_f rontend_b ytes_in_total	ClusterName , Namespace , Service	Alle Versionen von HAProxy Ingress
haproxy_f rontend_b ytes_out_total	ClusterName , Namespace , Service	Alle Versionen von HAProxy Ingress
haproxy_f rontend_c onnection s_total	ClusterName , Namespace , Service	Alle Versionen von HAProxy Ingress

Metrikname	Dimensionen	Verfügbarkeit
haproxy_frontend_current_sessions	ClusterName , Namespace , Service	Alle Versionen von HAProxy Ingress
haproxy_frontend_http_requests_total	ClusterName , Namespace , Service	Alle Versionen von HAProxy Ingress
haproxy_frontend_http_responses_total	ClusterName , Namespace , Service, Code, Frontend	Alle Versionen von HAProxy Ingress
haproxy_frontend_request_errors_total	ClusterName , Namespace , Service	Alle Versionen von HAProxy Ingress
haproxy_frontend_requests_denied_total	ClusterName , Namespace , Service	Alle Versionen von HAProxy Ingress

Note

Die Werte der code-Dimension können 1xx, 2xx, 3xx, 4xx, 5xx oder other sein.
Die Werte der backend-Dimension können wie folgt sein:

- http-default-backend, http-shared-backend, oder httpsback-shared-backend für HAProxy Ingress Version 0.0.27 oder früher.
- _default_backend für HAProxy Ingress Versionen höher als 0.0.27.

Die Werte der frontend-Dimension können wie folgt sein:

- `httpfront-default-backend`, `httpfront-shared-frontend`, oder `httpfronts` für HAProxy Ingress Version 0.0.27 oder früher.
- `_front_http` oder `_front_https` für HAProxy Ingress Versionen höher als 0.0.27.

Anzeigen Ihrer Prometheus-Metriken

Sie können alle Ihre Prometheus-Metriken überwachen und Alarme erhalten, einschließlich der kuratierten voraggregierten Metriken von App Mesh, NGINX, Java/JMX, Memcached und HAProxy und jedem anderen manuell konfigurierten Prometheus-Exporter, den Sie möglicherweise hinzugefügt haben. Weitere Informationen zum Erfassen von Metriken von anderen Prometheus-Exportern finden Sie unter Tutorial zum Hinzufügen eines neuen Prometheus-Scrape-Ziels: Prometheus-API-Server-Metriken.

In der CloudWatch Konsole stellt Container Insights die folgenden vorgefertigten Berichte bereit:

- Für Amazon-EKS- und Kubernetes-Cluster gibt es vordefinierte Berichte für App Mesh, NGINX, HAPROXY, Memcached und Java/JMX.
- Für Amazon-ECS-Cluster gibt es vordefinierte Berichte für App Mesh und Java/JMX.

Container Insights stellt auch benutzerdefinierte Dashboards für jede der Workloads bereit, aus denen Container Insights kuratierte Metriken sammelt. Sie können diese Dashboards von herunterladen GitHub

So zeigen Sie alle Prometheus-Metriken an:

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie in der Liste der Namespaces /Prometheus oder ContainerInsightsECS/ /Prometheus aus. ContainerInsights
4. Wählen Sie einen der Dimensionssätze in der folgenden Liste aus. Markieren Sie dann das Kontrollkästchen neben den Metriken, die Sie anzeigen möchten.

So zeigen Sie vorgefertigte Berichte zu Ihren Prometheus-Metriken an:

1. CloudWatch Öffnen Sie die Konsole unter https://console.aws.amazon.com/cloudwatch/.

2. Wählen Sie im Navigationsbereich Performance Monitoring (Leistungsüberwachung) aus.
3. Wählen Sie im Dropdown-Feld oben auf der Seite eine der Prometheus-Optionen aus.

   Wählen Sie im anderen Dropdown-Feld einen Cluster aus, der angezeigt werden soll.

Wir haben auch benutzerdefinierte Dashboards für NGINX, App Mesh, Memcached, HAProxy und Java/JMX bereitgestellt.

So verwenden Sie ein von Amazon bereitgestelltes benutzerdefiniertes Dashboard:

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Dashboards aus.
3. Klicken Sie auf Create Dashboard (Dashboard erstellen). Geben Sie einen Namen für das neue Dashboard ein und wählen Sie Create dashboard (Dashboard erstellen).
4. Wählen Sie unter Add to this dashboard (Zu diesem Dashboard hinzufügen) die Option Cancel (Abbrechen).
5. Wählen Sie Actions (Aktionen), View/edit source (Quelle anzeigen/bearbeiten).
6. Laden Sie eine der folgenden JSON-Dateien herunter:

   - Benutzerdefinierte NGINX-Dashboard-Quelle auf Github.
   - Quelle für benutzerdefiniertes App-Mesh-Dashboard auf Github.
   - Memcached benutzerdefinierte Dashboard-Quelle auf Github
   - HAProxy-Ingress benutzerdefinierte Dashboard-Quelle auf Github
   - Benutzerdefinierte Java/JMX-Dashboard-Quelle auf Github.
7. Öffnen Sie die heruntergeladene JSON-Datei mit einem Texteditor und nehmen Sie die folgenden Änderungen vor:

   - Ersetzen Sie alle `{{YOUR_CLUSTER_NAME}}`-Zeichenfolgen durch den genauen Namen Ihres Clusters. Stellen Sie sicher, dass keine Leerzeichen vor oder nach dem Text hinzugefügt werden.
   - Ersetzen Sie alle `{{YOUR_REGION}}` Zeichenketten durch die AWS Region, in der Ihr Cluster ausgeführt wird. Zum Beispiel **`us-west-1`**. Stellen Sie sicher, dass keine Leerzeichen vor oder nach dem Text hinzugefügt werden.
   - Ersetzen Sie alle `{{YOUR_NAMESPACE}}`-Zeichenfolgen durch den genauen Namespace Ihres Workloads.

- Ersetzen Sie alle `{{YOUR_SERVICE_NAME}}`-Zeichenfolgen durch den genauen Servicenamen Ihres Workloads. Zum Beispiel, **`haproxy-haproxy-ingress-controller-metrics`**

8. Kopieren Sie den gesamten JSON-Blob und fügen Sie ihn in das Textfeld in der CloudWatch Konsole ein. Dabei wird der Inhalt des Felds ersetzt.
9. Wählen Sie Update (Aktualisieren), Save dashboard (Dashboard speichern).

Fehlerbehebung für Prometheus-Metriken

Dieser Abschnitt enthält Hilfe zur Fehlerbehebung bei der Einrichtung Ihrer Prometheus-Metriken.

Themen

- Fehlerbehebung bei Prometheus-Metriken in Amazon ECS
- Fehlerbehebung bei Prometheus-Metriken in Amazon-EKS- und Kubernetes-Clustern

Fehlerbehebung bei Prometheus-Metriken in Amazon ECS

Dieser Abschnitt bietet Hilfe bei der Fehlerbehebung bei der Einrichtung von Prometheus-Metriken in Amazon-ECS-Clustern.

Ich sehe keine an Logs gesendeten Prometheus-Metriken CloudWatch

Die Prometheus-Metriken sollten als Protokollereignisse in die Protokollgruppe /aws/ecs/containerinsights/cluster-name/Prometheus aufgenommen werden. Wenn die Protokollgruppe nicht erstellt wurde oder die Prometheus-Metriken nicht an die Protokollgruppe gesendet werden, müssen Sie zunächst überprüfen, ob die Prometheus-Ziele erfolgreich vom Agenten erkannt wurden. CloudWatch Überprüfen Sie als Nächstes die Sicherheitsgruppen und die Berechtigungseinstellungen des Agenten. CloudWatch Die folgenden Schritte führen Sie zum Debuggen.

Schritt 1: Aktivieren Sie den CloudWatch Agenten-Debugging-Modus

Ändern Sie zunächst den CloudWatch Agenten in den Debug-Modus, indem Sie Ihrer AWS CloudFormation Vorlagendatei die folgenden fetten Zeilen hinzufügen, `cwagent-ecs-prometheus-metric-for-bridge-host.yaml` oder. `cwagent-ecs-prometheus-metric-for-awsvpc.yaml` Speichern Sie dann die Datei.

```
cwagentconfig.json: |
```

```
    {
      "agent": {
        "debug": true
      },
      "logs": {
        "metrics_collected": {
```

Erstellen Sie ein neues AWS CloudFormation Changeset für den vorhandenen Stack. Setzen Sie andere Parameter im Changeset auf dieselben Werte wie in Ihrem vorhandenen Stack. AWS CloudFormation Das folgende Beispiel bezieht sich auf einen CloudWatch Agenten, der in einem Amazon ECS-Cluster unter Verwendung des EC2-Starttyps und des Bridge-Netzwerkmodus installiert ist.

```
ECS_NETWORK_MODE=bridge
 CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name
NEW_CHANGESET_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=$ECS_EXECUTION_ROLE_NAME
 \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
    --change-set-name $NEW_CHANGESET_NAME
```

Gehen Sie zur AWS CloudFormation Konsole, um das neue Changeset zu überprüfen,. $NEW_CHANGESET_NAME Es sollte eine Änderung an der CW AgentConfig SSMParameter-Ressource vorgenommen werden. Führen Sie das Changeset aus und starten Sie die CloudWatch Agententask neu, indem Sie die folgenden Befehle eingeben.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service your_service_name_here \
--region $AWS_REGION
```

Warten Sie etwa 10 Sekunden und geben Sie den folgenden Befehl ein.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service your_service_name_here \
--region $AWS_REGION
```

Schritt 2: Prüfen der ECS-Serviceerkennungsprotokolle

Die ECS-Aufgabendefinition des CloudWatch Agenten aktiviert die Protokolle standardmäßig im folgenden Abschnitt. Die Protokolle werden an CloudWatch Logs in der Protokollgruppe ecs-cwagent-prometheus/ecs/ gesendet.

```
LogConfiguration:
  LogDriver: awslogs
    Options:
      awslogs-create-group: 'True'
      awslogs-group: "/ecs/ecs-cwagent-prometheus"
      awslogs-region: !Ref AWS::Region
      awslogs-stream-prefix: !Sub 'ecs-${ECSLaunchType}-awsvpc'
```

Filtern Sie die Protokolle nach der Zeichenfolge `ECS_SD_Stats`, um die Metriken im Zusammenhang mit der ECS-Serviceerkennung abzurufen, wie im folgenden Beispiel gezeigt.

```
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeContainerInstances: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeInstancesRequest: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeTaskDefinition: 2
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeTasks: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_ListTasks: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: Exporter_DiscoveredTargetCount: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Get_EC2MetaData: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Get_TaskDefinition: 2
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Size_ContainerInstance: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Size_TaskDefinition: 2
2020-09-1T01:53:14Z D! ECS_SD_Stats: Latency: 43.399783ms
```

Die Bedeutung jeder Metrik für einen bestimmten ECS-Serviceerkennungszyklus lautet wie folgt:

- AWSCLI_DescribeContainerInstances— die Anzahl der getätigten `ECS::DescribeContainerInstances` API-Aufrufe.

- AWSCLI_DescribeInstancesRequest— die Anzahl der `ECS::DescribeInstancesRequest` getätigten API-Aufrufe.
- AWSCLI_DescribeTaskDefinition— die Anzahl der `ECS::DescribeTaskDefinition` getätigten API-Aufrufe.
- AWSCLI_DescribeTasks— die Anzahl der `ECS::DescribeTasks` getätigten API-Aufrufe.
- AWSCLI_ListTasks— die Anzahl der `ECS::ListTasks` getätigten API-Aufrufe.
- ExporterDiscoveredTargetCount— die Anzahl der Prometheus-Ziele, die entdeckt und erfolgreich in die Zielergebnisdatei im Container exportiert wurden.
- lruCache_Get_EC2 MetaData — die Häufigkeit, mit der Metadaten der Container-Instance aus dem Cache abgerufen wurden.
- lruCache_get_ TaskDefinition — die Häufigkeit, mit der Metadaten der ECS-Aufgabendefinition aus dem Cache abgerufen wurden.
- lruCache_Size_ ContainerInstance — die Anzahl der Metadaten einer eindeutigen Container-Instance, die im Speicher zwischengespeichert wurden.
- lruCache_Size_ TaskDefinition — die Anzahl der eindeutigen ECS-Aufgabendefinitionen, die im Speicher zwischengespeichert sind.
- Latency – wie lange der Service-Discovery-Zyklus dauert.

Überprüfen Sie den Wert von `ExporterDiscoveredTargetCount`, um zu sehen, ob die erkannten Prometheus-Ziele Ihren Erwartungen entsprechen. Wenn nicht, sind die möglichen Gründe wie folgt:

- Die Konfiguration der ECS-Serviceerkennung stimmt möglicherweise nicht mit der Einstellung Ihrer Anwendung überein. Für die auf Docker-Labels basierende Diensterkennung ist für Ihre Zielcontainer möglicherweise nicht das erforderliche Docker-Label im CloudWatch Agenten konfiguriert, um sie auto zu erkennen. Für die auf regulären Ausdrücken basierende ARN-Diensterkennung für die ECS-Aufgabendefinition stimmt die Regex-Einstellung im CloudWatch Agenten möglicherweise nicht mit der Aufgabendefinition Ihrer Anwendung überein.
- Die ECS-Aufgabenrolle des CloudWatch Agenten ist möglicherweise nicht berechtigt, die Metadaten von ECS-Aufgaben abzurufen. Vergewissern Sie sich, dass dem CloudWatch Agenten die folgenden Nur-Lese-Berechtigungen gewährt wurden:
  - `ec2:DescribeInstances`
  - `ecs:ListTasks`
  - `ecs:DescribeContainerInstances`
  - `ecs:DescribeTasks`

- `ecs:DescribeTaskDefinition`

Schritt 3: Überprüfen Sie die Netzwerkverbindung und die ECS-Aufgabenrollenrichtlinie

Wenn immer noch keine Protokollereignisse an die CloudWatch Log-Zielgruppe Logs gesendet werden, obwohl der Wert von `Exporter_DiscoveredTargetCount` angibt, dass es entdeckte Prometheus-Ziele gibt, kann dies folgende Ursachen haben:

- Der CloudWatch Agent kann möglicherweise keine Verbindung zu den Prometheus-Zielports herstellen. Überprüfen Sie die Sicherheitsgruppeneinstellung hinter dem CloudWatch Agenten. Die private IP sollte es dem CloudWatch Agenten ermöglichen, eine Verbindung zu den Prometheus-Exporter-Ports herzustellen.
- Die ECS-Aufgabenrolle des CloudWatch Agenten verfügt möglicherweise nicht über die verwaltete Richtlinie. CloudWatchAgentServerPolicy Die ECS-Aufgabenrolle des CloudWatch Agenten muss über diese Richtlinie verfügen, um die Prometheus-Metriken als Protokollereignisse senden zu können. Wenn Sie die AWS CloudFormation Beispielvorlage verwendet haben, um die IAM-Rollen automatisch zu erstellen, werden sowohl der ECS-Aufgabenrolle als auch der ECS-Ausführungsrolle die geringste Berechtigung zur Durchführung der Prometheus-Überwachung gewährt.

Fehlerbehebung bei Prometheus-Metriken in Amazon-EKS- und Kubernetes-Clustern

Dieser Abschnitt bietet Hilfe bei der Fehlerbehebung bei der Einrichtung von Prometheus-Metriken in Amazon EKS- und Kubernetes-Clustern.

Schritte zur Problembehandlung bei Amazon EKS

Geben Sie den folgenden Befehl ein, um zu bestätigen, dass der CloudWatch Agent ausgeführt wird.

```
kubectl get pod -n amazon-cloudwatch
```

Die Ausgabe sollte eine Zeile mit `cwagent-prometheus-`*`id`* in der Spalte `NAME` und `Running` in der Spalte `STATUS column.` sein.

Geben Sie den folgenden Befehl ein, um Details zum laufenden Pod anzuzeigen. Ersetzen Sie *`pod-name`* durch den vollständigen Namen Ihres Pods, dessen Name mit `cw-agent-prometheus` beginnt.

```
kubectl describe pod pod-name -n amazon-cloudwatch
```

Wenn Sie CloudWatch Container Insights installiert haben, können Sie Logs Insights verwenden, um die CloudWatch Logs von dem CloudWatch Agenten abzufragen, der die Prometheus-Metriken sammelt.

So fragen Sie die Anwendungsprotokolle ab:

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich CloudWatch Logs Insights aus.
3. Wählen Sie die Protokollgruppe für die Anwendungsprotokolle, /aws/containerinsights/***cluster-name***/application
4. Ersetzen Sie den Suchabfrageausdruck durch die folgende Abfrage und wählen Sie Run query (Abfrage ausführen)

   ```
   fields ispresent(kubernetes.pod_name) as haskubernetes_pod_name, stream,
    kubernetes.pod_name, log |
   filter haskubernetes_pod_name and kubernetes.pod_name like /cwagent-prometheus
   ```

Sie können auch bestätigen, dass Prometheus-Metriken und -Metadaten als CloudWatch Logs-Ereignisse aufgenommen werden.

So bestätigen Sie, dass Prometheus-Daten aufgenommen werden:

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich CloudWatch Logs Insights aus.
3. Wählen Sie die /aws/containerinsights/***cluster-name***/prometheus
4. Ersetzen Sie den Suchabfrageausdruck durch die folgende Abfrage und wählen Sie Run query (Abfrage ausführen)

   ```
   fields @timestamp, @message | sort @timestamp desc | limit 20
   ```

Protokollierung von gelöschten Prometheus-Metriken

In dieser Version werden keine Prometheus-Metriken des Histogrammtyps erfasst. Sie können den CloudWatch Agenten verwenden, um zu überprüfen, ob Prometheus-Metriken gelöscht wurden, da

es sich um Histogramm-Metriken handelt. Sie können auch eine Liste der ersten 500 Prometheus-Metriken protokollieren, die gelöscht und nicht an sie gesendet wurden, CloudWatch da es sich um Histogramm-Metriken handelt.

Um festzustellen, ob Metriken gelöscht werden, geben Sie den folgenden Befehl ein:

```
kubectl logs -l "app=cwagent-prometheus" -n amazon-cloudwatch --tail=-1
```

Wenn Metriken gelöscht werden, werden die folgenden Zeilen in der `/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log`-Datei angezeigt.

```
I! Drop Prometheus metrics with unsupported types. Only Gauge, Counter and Summary are
 supported.
I! Please enable CWAgent debug mode to view the first 500 dropped metrics
```

Wenn Sie diese Zeilen sehen und wissen möchten, welche Metriken gelöscht werden, führen Sie die folgenden Schritte aus.

So protokollieren Sie eine Liste der gelöschten Prometheus-Metriken:

1. Versetzen Sie den CloudWatch Agenten in den Debug-Modus, indem Sie Ihrer OR-Datei die folgenden fetten Zeilen hinzufügen, und `prometheus-eks.yaml` speichern Sie die `prometheus-k8s.yaml` Datei.

   ```
   {
         "agent": {
           "debug": true
         },
   ```

   Dieser Abschnitt der Datei sollte dann wie folgt aussehen:

   ```
   cwagentconfig.json: |
       {
         "agent": {
           "debug": true
         },
         "logs": {
           "metrics_collected": {
   ```

2. Installieren Sie den CloudWatch Agenten erneut, um den Debug-Modus zu aktivieren, indem Sie die folgenden Befehle eingeben:

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
kubectl apply -f prometheus.yaml
```

   Die gelöschten Metriken werden im CloudWatch Agent-Pod protokolliert.

3. Geben Sie den folgenden Befehl ein, um die Protokolle aus dem CloudWatch Agenten-Pod abzurufen:

```
kubectl logs -l "app=cwagent-prometheus" -n amazon-cloudwatch --tail=-1
```

   *Oder, wenn Sie Container Insights Fluentd Logging installiert haben, werden die Protokolle auch in der Logs-Protokollgruppe **/aws/containerinsights/ CloudWatch cluster_name /application** gespeichert.*

   Um diese Protokolle abzufragen, können Sie die Schritte zum Abfragen der Anwendungsprotokolle in Schritte zur Problembehandlung bei Amazon EKS befolgen.

Wo werden die Prometheus-Metriken als CloudWatch Log-Log-Ereignisse aufgenommen?

Der CloudWatch Agent erstellt einen Protokollstream für jede Prometheus-Scrape-Job-Konfiguration. Beispiel: In den Dateien `prometheus-eks.yaml` und `prometheus-k8s.yaml` führt die Zeile `job_name: 'kubernetes-pod-appmesh-envoy'` das Scraping der App-Mesh-Metriken durch. Das Prometheus-Ziel ist definiert als `kubernetes-pod-appmesh-envoy`. Daher werden alle App Mesh Prometheus-Metriken als CloudWatch Logs-Ereignisse im Log-Stream kubernetes-pod-appmesh-envoyunter der Protokollgruppe /aws/containerinsights/cluster-name/Prometheus aufgenommen.

Ich sehe keine Amazon EKS- oder Kubernetes Prometheus-Metriken in den Metriken CloudWatch

Stellen Sie zunächst sicher, dass die Prometheus-Metriken als Protokollereignisse in der Protokollgruppe /aws/containerinsights/cluster-name/Prometheus aufgenommen werden. Verwenden Sie die Informationen in Wo werden die Prometheus-Metriken als CloudWatch Log-Log-Ereignisse aufgenommen?, um den Zielprotokoll-Stream zu überprüfen. Wenn der Protokoll-Stream nicht erstellt wird oder keine neuen Protokollereignisse im Protokoll-Stream vorhanden sind, überprüfen Sie Folgendes:

- Überprüfen Sie, ob die Prometheus-Metrik-Exporter-Endpunkte korrekt eingerichtet sind
- Überprüfen Sie, ob die Prometheus-Scraping-Konfigurationen im `config map: cwagent-prometheus` Abschnitt der CloudWatch Agenten-YAML-Datei korrekt sind. Die Konfiguration sollte die gleiche sein wie in einer Prometheus-Konfigurationsdatei. Weitere Informationen finden Sie unter <scrape_config> in der Prometheus-Dokumentation.

Wenn die Prometheus-Metriken korrekt als Protokollereignisse aufgenommen wurden, überprüfen Sie, ob die Einstellungen für das eingebettete Metrikformat zu den Protokollereignissen hinzugefügt wurden, um die Metriken zu generieren. CloudWatch

```
"CloudWatchMetrics":[
   {
      "Metrics":[
         {
            "Name":"envoy_http_downstream_cx_destroy_remote_active_rq"
         }
      ],
      "Dimensions":[
         [
            "ClusterName",
            "Namespace"
         ]
      ],
      "Namespace":"ContainerInsights/Prometheus"
   }
],
```

Weitere Hinweise zum eingebetteten Metrik-Format finden Sie unter Spezifikation: Eingebettetes Metrikformat .

Wenn die Protokollereignisse kein eingebettetes metrisches Format enthalten, überprüfen Sie, ob der `metric_declaration` Abschnitt im Abschnitt der YAML-Datei für die `config map: prometheus-cwagentconfig` CloudWatch Agenteninstallation korrekt konfiguriert ist. Weitere Informationen finden Sie unter Tutorial zum Hinzufügen eines neuen Prometheus-Scrape-Ziels: Prometheus-API-Server-Metriken.

Integration in Application Insights

Amazon CloudWatch Application Insights unterstützt Sie bei der Überwachung Ihrer Anwendungen und identifiziert und richtet wichtige Kennzahlen, Protokolle und Alarme für Ihre

Anwendungsressourcen und Ihren Technologie-Stack ein. Weitere Informationen finden Sie unter Einblicke in CloudWatch Amazon-Anwendungen.

Sie können Application Insights aktivieren, um zusätzliche Daten aus Ihren containerisierten Anwendungen und Microservices zu sammeln. Wenn noch nicht getan, können Sie es unter der Leistungsansicht auf dem Container-Insights-Dashboard tun (mit Auto-configure Application Insights (Application Insights automatisch konfigurieren)).

Wenn Sie CloudWatch Application Insights bereits für die Überwachung Ihrer containerisierten Anwendungen eingerichtet haben, wird das Application Insights-Dashboard unter dem Container Insights-Dashboard angezeigt.

Weitere Informationen zu Application Insights und containerisierten Anwendungen finden Sie unter Aktivieren von Application Insights zur Ressourcenüberwachung für Amazon ECS und Amazon EKS.

Ansehen von Amazon-ECS-Lebenszyklusereignissen in Container Insights

Sie können Amazon-ECS-Lebenszyklusereignisse in der Container-Insights-Konsole ansehen. Auf diese Weise können Sie Ihre Container-Metriken, Protokolle und Ereignisse in einer einzigen Ansicht korrelieren und Sie erhalten einen umfassenderen Einblick in die Funktionalität.

Zu den Ereignissen zählen Statusänderungsereignisse für Container-Instances, Änderungsereignisse für den Aufgabenstatus und Service-Aktionsereignisse. Sie werden automatisch von Amazon ECS an Amazon gesendet EventBridge und auch CloudWatch im Ereignisprotokollformat erfasst. Weitere Informationen zu diesen Ereignissen finden Sie unter Amazon-ECS-Ereignisse.

Für Amazon ECS Lifecycle-Ereignisse gelten die Standardpreise von Container Insights. Weitere Informationen finden Sie unter Amazon CloudWatch – Preise.

Um die Tabelle der Lebenszyklusereignisse zu konfigurieren und Regeln für einen Cluster zu erstellen, müssen Sie die Berechtigungen `events:PutRule`, `events:PutTargets` und `logs:CreateLogGroup` haben. Sie müssen außerdem sicherstellen, dass es eine Ressourcenrichtlinie gibt, die es ermöglicht, den Protokollstream EventBridge zu erstellen und Protokolle an Logs zu CloudWatch senden. Wenn diese Ressourcenrichtlinie nicht existiert, können Sie den folgenden Befehl eingeben, um sie zu erstellen:

```
aws --region region logs put-resource-policy --policy-name 'EventBridgeCloudWatchLogs'
 --policy-document '{
    "Statement": [
        {
```

```
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": ["events.amazonaws.com", "delivery.logs.amazonaws.com"]
            },
            "Resource": "arn:aws:logs:region:account-id:log-group:/aws/events/ecs/
containerinsights/*:*",
            "Sid": "TrustEventBridgeToStoreECSLifecycleLogEvents"
        }
    ],
    "Version": "2012-10-17"
}'
```

Sie können den folgenden Befehl verwenden, um zu überprüfen, ob Sie diese Richtlinie bereits haben, und um zu bestätigen, dass das Anhängen ordnungsgemäß funktioniert hat.

```
aws logs describe-resource-policies --region region --output json
```

Zum Anzeigen der Tabelle mit Lebenszyklusereignissen benötigen Sie die Berechtigungen `events:DescribeRule`, `events:ListTargetsByRule` und`logs:DescribeLogGroups`.

So zeigen Sie Amazon ECS-Lebenszyklusereignisse in der CloudWatch Container Insights-Konsole an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie Insights (Einblicke), Container Insights.
3. Wählen Sie Leistungs-Dashboards anzeigen aus.
4. Wählen Sie im nächsten Dropdown eine der folgenden Optionen: ECS Clusters (ECS-Cluster), ECS Services (ECS-Services) oder ECS Tasks (ECS-Aufgaben).
5. Wenn Sie im vorherigen Schritt ECS Services (ECS-Services) oder ECS Tasks (ECS-Aufgaben) ausgewählt haben, wählen Sie die Registerkarte Lifecycle events (Lebenszyklusereignisse) aus.
6. Wenn Sie unten auf der Seite die Option Lebenszyklusereignisse konfigurieren sehen, wählen Sie diese Option aus, um EventBridge Regeln für Ihren Cluster zu erstellen.

   Die Ereignisse werden unter den Container-Insights-Bereichen und über dem Abschnitt Application Insights angezeigt. Um weitere Analysen durchzuführen und zusätzliche

Visualisierungen für diese Ereignisse zu erstellen, wählen Sie View in Logs Insights (In Logs Insights anzeigen) in der Tabelle mit den Lebenszyklusereignissen aus.

Fehlerbehebung bei Container Insights

Die folgenden Abschnitte können hilfreich sein, wenn Sie Probleme mit Container Insights haben.

Fehlgeschlagene Bereitstellung auf Amazon EKS oder Kubernetes

Wenn der Agent nicht ordnungsgemäß auf einem Kubernetes-Cluster bereitgestellt wird, versuchen Sie Folgendes:

- Führen Sie den folgenden Befehl aus, um die Liste der Pods zu erhalten.

```
kubectl get pods -n amazon-cloudwatch
```

- Führen Sie den folgenden Befehl aus und überprüfen Sie die Ereignisse am unteren Rand der Ausgabe.

```
kubectl describe pod pod-name -n amazon-cloudwatch
```

- Führen Sie den folgenden Befehl aus, um die Protokolle zu überprüfen.

```
kubectl logs pod-name -n amazon-cloudwatch
```

Nicht autorisierte Panik: Kann keine cadvisor-Daten aus kubelet abrufen

Falls Ihre Bereitstellung mit der Fehlermeldung `Unauthorized panic: Cannot retrieve cadvisor data from kubelet` fehlschlägt, ist der Webhook-Autorisierungsmodus für kubelet möglicheweise nicht aktiviert. Dieser Modus ist für Container Insights erforderlich. Weitere Informationen finden Sie unter Überprüfen Sie die -Voraussetzungen..

Bereitstellen von Container Insights auf einem gelöschten und neu erstellten Cluster auf Amazon ECS

Wenn Sie einen vorhandenen Amazon-ECS-Cluster löschen, für den Container Insights nicht aktiviert ist, und Sie ihn mit demselben Namen neu erstellen, können Sie Container Insights für diesen neuen Cluster, bei der Neuerstellung, nicht aktivieren. Sie können es aktivieren, indem Sie ihn neu erstellen und dann den folgenden Befehl eingeben:

```
aws ecs update-cluster-settings --cluster myCICluster --settings
 name=containerInsights,value=enabled
```

„Ungültiger Endpunkt“-Fehler

Wenn eine Fehlermeldung ähnlich der folgenden angezeigt wird, stellen Sie sicher, dass Sie alle Platzhalter wie *cluster-name* und *region-name* in den von Ihnen verwendeten Befehlen durch die korrekten Informationen für Ihre Bereitstellung ersetzt haben.

```
"log": "2020-04-02T08:36:16Z E! cloudwatchlogs: code: InvalidEndpointURL, message:
 invalid endpoint uri, original error: &url.Error{Op:\"parse\", URL:\"https://
logs.{{region_name}}.amazonaws.com/\", Err:\"{\"}, &awserr.baseError{code:
\"InvalidEndpointURL\", message:\"invalid endpoint uri\", errs:[]error{(*url.Error)
(0xc0008723c0)}}\n",
```

Metriken erscheinen nicht in der Konsole

Wenn Sie in der keine Container Insights-Metriken sehen AWS Management Console, stellen Sie sicher, dass Sie die Einrichtung von Container Insights abgeschlossen haben. Metriken werden erst angezeigt, wenn Container Insights vollständig eingerichtet wurde. Weitere Informationen finden Sie unter Einrichten von Container Insights.

Pod-Metriken fehlen auf Amazon EKS oder Kubernetes nach dem Upgrade des Clusters

Dieser Abschnitt kann nützlich sein, wenn alle oder einige Pod-Metriken fehlen, nachdem Sie den CloudWatch Agenten als Daemonset auf einem neuen oder aktualisierten Cluster bereitgestellt haben, oder wenn Sie ein Fehlerprotokoll mit der Meldung sehen. `W! No pod metric collected`

Diese Fehler können durch Änderungen in der Container-Laufzeitumgebung verursacht werden, z. B. containerd oder den „docker systemd cgroup“-Treiber. Sie können dies normalerweise lösen, indem Sie Ihr Bereitstellungsmanifest aktualisieren, sodass der containerd-Socket vom Host in den Container gemountet wird. Sehen Sie sich das folgende Beispiel an:

```
# For full example see https://github.com/aws-samples/amazon-cloudwatch-container-
insights/blob/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/
container-insights-monitoring/cwagent/cwagent-daemonset.yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
```

```
  name: cloudwatch-agent
  namespace: amazon-cloudwatch
spec:
  template:
    spec:
      containers:
        - name: cloudwatch-agent
# ...
          # Don't change the mountPath
          volumeMounts:
# ...
            - name: dockersock
              mountPath: /var/run/docker.sock
              readOnly: true
            - name: varlibdocker
              mountPath: /var/lib/docker
              readOnly: true
            - name: containerdsock # NEW mount
              mountPath: /run/containerd/containerd.sock
              readOnly: true
# ...
      volumes:
# ...
        - name: dockersock
          hostPath:
            path: /var/run/docker.sock
        - name: varlibdocker
          hostPath:
            path: /var/lib/docker
        - name: containerdsock # NEW volume
          hostPath:
            path: /run/containerd/containerd.sock
```

Keine Pod-Metriken bei Verwendung von Bottlerocket für Amazon EKS

Bottlerocket ist ein Linux-basiertes Open-Source-Betriebssystem, das von AWS für das Ausführen von Container erstellt wurde.

Bottlerocket verwendet einen anderen `containerd`-Pfad auf dem Host, daher müssen Sie die Volumes an ihren Speicherort ändern. Wenn Sie dies nicht tun, wird in den Protokollen ein Fehler angezeigt, der `W! No pod metric collected` enthält. Sehen Sie sich das folgende -Beispiel an.

```
volumes:
```

```
  # ...
    - name: containerdsock
      hostPath:
        # path: /run/containerd/containerd.sock
        # bottlerocket does not mount containerd sock at normal place
        # https://github.com/bottlerocket-os/bottlerocket/
commit/91810c85b83ff4c3660b496e243ef8b55df0973b
        path: /run/dockershim.sock
```

Keine Container-Dateisystem-Metriken bei Verwendung der containerd-Laufzeitumgebung für Amazon EKS oder Kubernetes

Dies ist ein bekanntes Problem und wird von Community-Mitwirkenden bearbeitet. Weitere Informationen finden Sie unter Metrik zur Festplattennutzung für containerd und Container-Dateisystemmetriken werden von cadvisor für containerd on nicht unterstützt. GitHub

Unerwarteter Anstieg des Protokollvolumens durch den CloudWatch Agenten beim Sammeln von Prometheus-Metriken

Dies war eine Regression, die in Version 1.247347.6b250880 des Agenten eingeführt wurde. CloudWatch Diese Regression wurde bereits in neueren Versionen des Agenten behoben. Die Auswirkungen beschränkten sich auf Szenarien, in denen Kunden die Protokolle des CloudWatch Agenten selbst sammelten und auch Prometheus verwendeten. Weitere Informationen finden Sie unter [prometheus] Agent druckt alle gesammelten Metriken bei der Anmeldung aus. GitHub

Neueste Docker-Image, das in Versionshinweisen erwähnt wurde, die nicht von Dockerhub gefunden wurden

Wir aktualisieren die Versionshinweise und das Tag auf Github, bevor wir die eigentliche Veröffentlichung intern starten. In der Regel dauert es 1-2 Wochen, um das neueste Docker-Image in Registries zu sehen, nachdem wir die Versionsnummer auf Github stoßen. Es gibt keine nächtliche Veröffentlichung für das Agenten-Container-Image. CloudWatch Sie können das Image direkt aus dem Quellcode am folgenden Speicherort erstellen: https://github.com/aws/ amazon-cloudwatch-agent /tree/main//amazon-cloudwatch-container-insightscloudwatch-agent-dockerfile

CrashLoopBackoff Fehler auf dem Agenten CloudWatch

Wenn Sie einen `CrashLoopBackOff` Fehler für den CloudWatch Agenten sehen, stellen Sie sicher, dass Ihre IAM-Berechtigungen richtig eingestellt sind. Weitere Informationen finden Sie unter Überprüfen Sie die -Voraussetzungen..

CloudWatch Der Agent oder der Fluentd-Pod bleiben im Status „Ausstehend“ hängen

Wenn ein CloudWatch Agent oder ein Fluentd-Pod nicht mehr funktioniert `Pending` oder ein `FailedScheduling` Fehler auftritt, ermitteln Sie anhand der Anzahl der Kerne und der Menge an RAM, die von den Agenten benötigt werden, ob Ihre Knoten über genügend Rechenressourcen verfügen. Geben Sie den folgenden Befehl ein, um den Pod zu beschreiben:

```
kubectl describe pod cloudwatch-agent-85ppg -n amazon-cloudwatch
```

Erstellen Sie Ihr eigenes Docker-Image für CloudWatch Agenten

Sie können Ihr eigenes Docker-Image für CloudWatch Agenten erstellen, indem Sie auf das Dockerfile verweisen, das sich unter https://github.com/aws-samples/ /blob/latest/ /Dockerfile befindet. amazon-cloudwatch-container-insights cloudwatch-agent-dockerfile

Das Dockerfile unterstützt das Erstellen von Images mit mehreren Architekturen direkt mit `docker buildx`.

CloudWatch Bereitstellung anderer Agentenfunktionen in Ihren Containern

Mit dem CloudWatch Agenten können Sie zusätzliche Überwachungsfunktionen in Ihren Containern bereitstellen. Nachstehend sind einige dieser Features aufgeführt:

- Embedded Metric Format (Eingebettetes Metrikformat) – Für weitere Informationen vgl. Einbetten von Metriken in Protokollen.
- StatsD – Weitere Informationen finden Sie unter Abrufen benutzerdefinierter Metriken mit StatsD .

Anweisungen und die erforderlichen Dateien befinden sich GitHub an den folgenden Speicherorten:

- Weitere Informationen zu Amazon-ECS-Containern finden Sie unter Beispiele für Amazon-ECS-Aufgabendefinitionen basierend auf den Bereitstellungsmodi.
- Für Amazon-EKS- und Kubernetes-Container finden Sie unter Beispiele für Kubernetes-YAML-Dateien basierend auf den Bereitstellungsmodi.

Lambda Insights

CloudWatch Lambda Insights ist eine Überwachungs- und Fehlerbehebungslösung für serverlose Anwendungen, die auf ausgeführt werden. AWS Lambda Die Lösung erfasst, aggregiert und fasst

Metriken auf Systemebene zusammen, einschließlich CPU-Zeit, Arbeitsspeicher, Datenträger und Netzwerk. Sie erfasst, aggregiert und fasst Diagnoseinformationen wie Kaltstart und Lambda-Worker-Abschaltungen zusammen, um Probleme mit Ihren Lambda-Funktionen zu isolieren und schnell zu beheben.

Lambda Insights verwendet eine neue CloudWatch Lambda-Erweiterung, die als Lambda-Schicht bereitgestellt wird. Wenn Sie diese Erweiterung auf einer Lambda-Funktion installieren, sammelt sie Metriken auf Systemebene und gibt für jeden Aufruf dieser Lambda-Funktion ein einziges Leistungsprotokollereignis aus. CloudWatch verwendet eine eingebettete Metrikformatierung, um Metriken aus den Protokollereignissen zu extrahieren.

Weitere Informationen zu Lambda-Erweiterungen finden Sie unter AWS Lambda Erweiterungen verwenden. Weitere Hinweise zum eingebetteten Metrik-Format finden Sie unter Einbetten von Metriken in Protokollen.

Sie können Lambda Insights mit jeder Lambda-Funktion verwenden, die eine Lambda-Laufzeit verwendet, die Lambda-Erweiterungen unterstützt. Eine Liste dieser Laufzeiten finden Sie unter Lambda-Erweiterungen-API.

Preise

Für jede für Lambda Insights aktivierte Lambda-Funktion zahlen Sie nur für das, was Sie für Metriken und Protokolle tatsächlich nutzen. Ein Preisbeispiel finden Sie unter Amazon CloudWatch Pricing.

Die von der Lambda-Erweiterung verbrauchte Ausführungszeit wird Ihnen in Schritten von je 1 ms in Rechnung gestellt. Weitere Informationen zu den Preisen für Lambda finden Sie unter AWS Lambda - Preise.

Erste Schritte mit Lambda Insights

Um Lambda Insights für eine Lambda-Funktion zu aktivieren, können Sie einen Ein-Klick-Schalter in der Lambda-Konsole verwenden. Alternativ können Sie die AWS CLI, AWS CloudFormation, die AWS Serverless Application Model CLI oder die verwenden AWS Cloud Development Kit (AWS CDK).

Die folgenden Abschnitte enthalten detaillierte Anweisungen zum Ausführen dieser Schritte.

Themen

- Verwenden der Konsole zum Aktivieren von Lambda Insights für eine vorhandene Lambda-Funktion
- Verwenden von AWS CLI , um Lambda Insights für eine bestehende Lambda-Funktion zu aktivieren
- Verwenden der AWS SAM CLI zur Aktivierung von Lambda Insights für eine bestehende Lambda-Funktion
- Wird verwendet AWS CloudFormation , um Lambda Insights für eine bestehende Lambda-Funktion zu aktivieren
- Verwenden von AWS CDK , um Lambda Insights für eine bestehende Lambda-Funktion zu aktivieren
- Verwenden des Serverless-Frameworks zum Aktivieren von Lambda Insights für eine vorhandene Lambda-Funktion
- Aktivieren von Lambda Insights für eine Lambda-Container-Image-Bereitstellung

Verfügbare Versionen der Lambda-Insights-Erweiterung

In diesem Abschnitt werden die Versionen der Lambda Insights-Erweiterung und die ARNs aufgeführt, die für diese Erweiterungen in jeder AWS Region verwendet werden sollen.

Themen

- x86-64-Plattformen
- ARM64-Plattformen

x86-64-Plattformen

In diesem Abschnitt werden die Versionen der Lambda Insights-Erweiterung für x86-64-Plattformen sowie die ARNs aufgeführt, die für diese Erweiterungen in jeder Region verwendet werden sollen. AWS

> **Important**
>
> Lambda Insights-Erweiterungen 1.0.317.0 und höher unterstützen Amazon Linux 1 nicht.

1.0.317.0

Version 1.0.317.0 beinhaltet die Entfernung der Unterstützung für die Amazon Linux 1-Plattform und Bugfixes. Sie beinhaltet auch Unterstützung für Regionen. AWS GovCloud (US)

ARNs für Version 1.0.317.0

In der folgenden Tabelle sind die ARNs aufgeführt, die für diese Version der Erweiterung in jeder AWS Region verwendet werden können, in der sie verfügbar ist.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:52`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:52`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:52`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:52`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:43`
Asien-Pazifik (Hongkong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:43`
Asien-Pazifik (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:25`
Asien-Pazifik (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:29`
Asien-Pazifik (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:20`

Region	ARN
Asien-Pazifik (Mumbai)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:50
Asia Pacific (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:33
Asia Pacific (Seoul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:51
Asien-Pazifik (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:52
Asien-Pazifik (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:52
Asien-Pazifik (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:79
Canada (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:51
Kanada West (Calgary)	arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:12
China (Peking)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:42
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:42
Europe (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:52
Europa (Irland)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:52
Europa (London)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:52

Region	ARN
Europa (Milan)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:43`
Europa (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:51`
Europa (Spain)	`arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:27`
Europa (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:49`
Europa (Zürich)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:26`
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:20`
Naher Osten (Bahrain)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:43`
Naher Osten (VAE)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:26`
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:51`
AWS GovCloud (US-Ost)	`arn:aws-us-gov:lambda:us-gov-east-1:122132214140:layer:LambdaInsightsExtension:19`
AWS GovCloud (US-West)	`arn:aws-us-gov:lambda:us-gov-west-1:751350123760:layer:LambdaInsightsExtension:19`

1.0.295.0

Version 1.0.295.0 enthält Abhängigkeitsupdates für alle kompatiblen Laufzeiten.

ARNs für Version 1.0.295.0

In der folgenden Tabelle sind die ARNs aufgeführt, die für diese Version der Erweiterung in jeder AWS Region verwendet werden können, in der sie verfügbar ist.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:51`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:51`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:51`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:51`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:42`
Asien-Pazifik (Hongkong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:42`
Asien-Pazifik (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:24`
Asien-Pazifik (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:28`
Asien-Pazifik (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:19`
Asien-Pazifik (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:49`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:32`

Region	ARN
Asia Pacific (Seoul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:50
Asien-Pazifik (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:51
Asien-Pazifik (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:51
Asien-Pazifik (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:78
Canada (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:50
Kanada West (Calgary)	arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:11
China (Peking)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:41
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:41
Europe (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:51
Europa (Irland)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:51
Europa (London)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:51
Europa (Milan)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:42
Europa (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:50

Region	ARN
Europa (Spain)	`arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:26`
Europa (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:48`
Europa (Zürich)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:25`
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:19`
Naher Osten (Bahrain)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:42`
Naher Osten (VAE)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:25`
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:50`

1.0.275.0

Version 1.0.275.0 enthält wichtige Abhängigkeitsupdates für alle kompatiblen Laufzeiten.

ARNs für Version 1.0.275.0

In der folgenden Tabelle sind die ARNs aufgeführt, die für diese Version der Erweiterung in jeder AWS Region verwendet werden können, in der sie verfügbar ist.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:49`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:49`

Region	ARN
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:49`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:49`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:40`
Asien-Pazifik (Hongkong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:40`
Asien-Pazifik (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:22`
Asien-Pazifik (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:26`
Asien-Pazifik (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:17`
Asien-Pazifik (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:47`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:30`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:48`
Asien-Pazifik (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:49`
Asien-Pazifik (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:49`
Asien-Pazifik (Tokio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:76`

Region	ARN
Canada (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:48
Kanada West (Calgary)	arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:9
China (Peking)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:39
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:39
Europe (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:49
Europa (Irland)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:49
Europa (London)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:49
Europa (Milan)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:40
Europa (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:48
Europa (Spain)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:24
Europa (Stockholm)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:46
Europa (Zürich)	arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:23
Israel (Tel Aviv)	arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:17

Region	ARN
Naher Osten (Bahrain)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:40`
Naher Osten (VAE)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:23`
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:48`

1.0.273.0

Version 1.0.273.0 enthält wichtige Bugfixes für alle kompatiblen Laufzeiten und fügt Unterstützung für Canada West (Calgary) hinzu.

ARNs für Version 1.0.273.0

In der folgenden Tabelle sind die ARNs aufgeführt, die für diese Version der Erweiterung in jeder AWS Region verwendet werden können, in der sie verfügbar ist.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:45`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:45`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:45`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:45`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:35`

Region	ARN
Asien-Pazifik (Hongkong)	arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:35
Asien-Pazifik (Hyderabad)	arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:17
Asien-Pazifik (Jakarta)	arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:21
Asien-Pazifik (Melbourne)	arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:12
Asien-Pazifik (Mumbai)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:43
Asia Pacific (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:26
Asia Pacific (Seoul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:44
Asien-Pazifik (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:45
Asien-Pazifik (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:45
Asien-Pazifik (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:72
Canada (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:44
Kanada West (Calgary)	arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:4
China (Peking)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:36

Region	ARN
China (Ningxia);	`arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:36`
Europe (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:45`
Europa (Irland)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:45`
Europa (London)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:45`
Europa (Milan)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:35`
Europa (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:44`
Europa (Spain)	`arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:19`
Europa (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:42`
Europa (Zürich)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:17`
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:12`
Naher Osten (Bahrain)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:35`
Naher Osten (VAE)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:18`
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:44`

1.0.229.0

Version 1.0.229.0 enthält wichtige Fehlerbehebungen für alle kompatiblen Laufzeiten und bietet neue Unterstützung für die Region Israel (Tel Aviv).

ARNs für Version 1.0.229.0

In der folgenden Tabelle sind die ARNs aufgeführt, die für diese Version der Erweiterung in jeder AWS Region verwendet werden sollen, in der sie verfügbar ist.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:38`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:38`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:38`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:38`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:28`
Asien-Pazifik (Hongkong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:28`
Asien-Pazifik (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:10`
Asien-Pazifik (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:14`
Asien-Pazifik (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:5`

Region	ARN
Asien-Pazifik (Mumbai)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:36
Asia Pacific (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:19
Asia Pacific (Seoul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:37
Asien-Pazifik (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:38
Asien-Pazifik (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:38
Asien-Pazifik (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:60
Canada (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:37
China (Peking)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:29
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:29
Europe (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:38
Europa (Irland)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:38
Europa (London)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:38
Europa (Milan)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:28

Region	ARN
Europa (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:37
Europa (Spain)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:12
Europa (Stockholm)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:35
Europa (Zürich)	arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:11
Israel (Tel Aviv)	arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:5
Naher Osten (Bahrain)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:28
Naher Osten (VAE)	arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:11
Südamerika (São Paulo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:37

1.0.178.0

Version 1.0.178.0 bietet Unterstützung für die folgenden Regionen. AWS

- Asien-Pazifik (Hyderabad)
- Asien-Pazifik (Jakarta)
- Europa (Spain)
- Europa (Zürich)
- Naher Osten (VAE)

ARNs für Version 1.0.178.0

In der folgenden Tabelle sind die ARNs aufgeführt, die für diese Version der Erweiterung in jeder AWS Region verwendet werden sollen, in der sie verfügbar ist.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:35`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:33`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:33`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:33`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:25`
Asien-Pazifik (Hongkong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:25`
Asien-Pazifik (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:8`
Asien-Pazifik (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:11`
Asien-Pazifik (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:31`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:2`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:32`

Region	ARN
Asien-Pazifik (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:33
Asien-Pazifik (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:33
Asien-Pazifik (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:50
Canada (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:32
China (Peking)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:26
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:26
Europe (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:35
Europa (Irland)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:33
Europa (London)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:33
Europa (Milan)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:25
Europa (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:32
Europa (Spain)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:10
Europa (Stockholm)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:30

Region	ARN
Europa (Zürich)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:7`
Naher Osten (Bahrain)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:25`
Naher Osten (VAE)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:9`
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:32`

1.0.143.0

Version 1.0.143.0 enthält Fehlerbehebungen in Kompatibilität mit Python 3.7 und Go 1.x. Die Python-3.6-Lambda-Laufzeit ist veraltet. Weitere Informationen finden Sie unter Lambda-Laufzeiten.

ARNs für Version 1.0.143.0

In der folgenden Tabelle sind die ARNs aufgeführt, die für diese Version der Erweiterung in jeder AWS Region verwendet werden können, in der sie verfügbar ist.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:21`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:21`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:20`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:21`

Region	ARN
Africa (Cape Town)	arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:13
Asia Pacific (Hong Kong)	arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:13
Asia Pacific (Mumbai)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:21
Asia Pacific (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:2
Asia Pacific (Seoul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:20
Asien-Pazifik (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:21
Asien-Pazifik (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:21
Asien-Pazifik (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:32
Canada (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:20
China (Peking)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:14
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:14
Europe (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:21
Europa (Irland)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:21

Region	ARN
Europa (London)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:21`
Europa (Milan)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:13`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:20`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:20`
Middle East (Bahrain)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:13`
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:20`

1.0.135.0

Version 1.0.135.0 enthält Fehlerkorrekturen dafür, wie Lambda Insights die Nutzung von Festplatten- und Dateibeschreibungen sammelt und meldet. In früheren Versionen der Erweiterung meldete die `tmp_free`-Metrik den maximal freien Speicherplatz im `/tmp`-Verzeichnis während der Ausführung einer Funktion. Diese Version ändert die Metrik, um stattdessen den Mindestwert zu melden, was bei der Bewertung der Festplattennutzung nützlicher ist. Weitere Informationen zu Speicherkontingenten für `tmp`-Verzeichnisse finden Sie unter Lambda-Kontingente.

Version 1.0.135.0 meldet jetzt auch die Nutzung von Dateibeschreibungen (`fd_use` und `fd_max`) als prozessübergreifenden Maximalwert, anstatt die Betriebssystemebene zu melden.

ARNs für Version 1.0.135.0

In der folgenden Tabelle sind die ARNs aufgeführt, die für diese Version der Erweiterung in jeder AWS Region verwendet werden können, in der sie verfügbar ist.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:18`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:18`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:18`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:18`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:11`
Asia Pacific (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:11`
Asia Pacific (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:18`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:1`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:18`
Asien-Pazifik (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:18`
Asien-Pazifik (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:18`
Asien-Pazifik (Tokio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:25`

Region	ARN
Canada (Central)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:18`
China (Peking)	`arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:11`
China (Ningxia);	`arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:11`
Europe (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:18`
Europa (Irland)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:18`
Europa (London)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:18`
Europa (Milan)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:11`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:18`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:18`
Middle East (Bahrain)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:11`
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:18`

1.0.119.0

ARNs für Version 1.0.119.0

In der folgenden Tabelle sind die ARNs aufgeführt, die für diese Version der Erweiterung in jeder AWS Region verwendet werden sollen, in der sie verfügbar ist.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:16`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:16`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:16`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:16`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:9`
Asia Pacific (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:9`
Asien-Pazifik (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:16`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:16`
Asien-Pazifik (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:16`
Asien-Pazifik (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:16`
Asien-Pazifik (Tokio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:23`

Region	ARN
Canada (Central)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:16`
China (Peking)	`arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:9`
China (Ningxia);	`arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:9`
Europe (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:16`
Europa (Irland)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:16`
Europa (London)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:16`
Europa (Milan)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:9`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:16`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:16`
Middle East (Bahrain)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:9`
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:16`

1.0.98.0

Diese Version entfernt unnötige Protokollierung und behebt auch ein Problem mit den lokalen AWS Serverless Application Model -CLI-Aufrufen. Weitere Informationen zu diesem Problem finden Sie unter Hinzufügen von LambdaInsightsExtension Ergebnissen bei Timeout mit 'sam local invoke'.

ARNs für Version 1.0.98.0

In der folgenden Tabelle sind die ARNs aufgeführt, die für diese Version der Erweiterung in jeder AWS Region verwendet werden können, in der sie verfügbar ist.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:14`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:14`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:14`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:14`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:8`
Asia Pacific (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:8`
Asien-Pazifik (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:14`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:14`
Asien-Pazifik (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:14`

Region	ARN
Asien-Pazifik (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:14
Asien-Pazifik (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:14
Canada (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:14
China (Peking)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:8
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:8
Europe (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:14
Europa (Irland)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:14
Europa (London)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:14
Europa (Milan)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:8
Europe (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:14
Europe (Stockholm)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:14
Middle East (Bahrain)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:8
Südamerika (São Paulo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:14

1.0.89.0

Diese Version korrigiert den Zeitstempel des Leistungsereignisses, um immer den Beginn des Aufrufs der Funktion darzustellen.

ARNs für Version 1.0.89.0

In der folgenden Tabelle sind die ARNs aufgeführt, die für diese Version der Erweiterung in jeder AWS Region verwendet werden können, in der sie verfügbar ist.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:12`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:12`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:12`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:12`
Asia Pacific (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:12`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:12`
Asien-Pazifik (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:12`
Asien-Pazifik (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:12`
Asien-Pazifik (Tokio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:12`

Region	ARN
Canada (Central)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:12`
Europe (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:12`
Europa (Irland)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:12`
Europe (London)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:12`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:12`
Europa (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:12`
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:12`

1.0.86.0

Mit Version 1.0.54.0 der Erweiterung wurden Speichermetriken manchmal falsch gemeldet und waren manchmal höher als 100 %. Version 1.0.86.0 behebt das Problem der Speichermessung, indem dieselben Ereignisdaten wie Lambda-Plattformmetriken verwendet werden. Dies bedeutet, dass Sie eine dramatische Änderung der aufgezeichneten Speichermetrikwerte sehen können. Dies wird durch die Verwendung der neuen Lambda-Protokoll-API erreicht. Dies ermöglicht eine genauere Messung der Lambda-Sandbox-Speichernutzung. Beachten Sie jedoch, dass die Lambda-Protokoll-API keine Plattformberichtsereignisse liefern kann, wenn eine Funktions-Sandbox das Zeitlimit überschreitet und anschließend heruntergefahren wird. In diesem Fall. Lambda Insights kann die Aufruf-Metriken nicht aufzeichnen. Weitere Informationen zur Lambda-Protokoll-API finden Sie unter AWS -Lambda-Protokoll-API.

Neue Features in Version 1.0.86.0

- Verwendet die Lambda-Protokoll-API, um die Speichermetrik zu korrigieren. Dies behebt das vorherige Problem, bei dem Speicherstatistiken größer als 100 % waren.
- Wird `Init Duration` als neue Metrik eingeführt. CloudWatch
- Verwendet den Aufruf-ARN, um eine Versions-Dimension für Aliase und aufgerufene Versionen hinzuzufügen. Wenn Sie Lambda-Aliasse oder -Versionen verwenden, um inkrementelle Bereitstellungen (z. B. blau-grüne Bereitstellungen) zu erreichen, können Sie Ihre Metriken basierend auf dem aufgerufenen Alias anzeigen. Die Version-Dimension wird nicht angewendet, wenn die Funktion keinen Alias oder keine Version verwendet. Weitere Informationen erhalten Sie unter Lambda-Funktionen-Aliase.
- Fügt den Leistungsereignissen ein `billed_mb_ms field` hinzu, um die Kosten pro Aufruf anzuzeigen. Hierbei werden keine Kosten berücksichtigt, die mit bereitgestellter Parallelität verbunden sind.
- Fügt den Leistungsereignissen `billed_duration`- und `duration`-Felder hinzu.

ARNs für Version 1.0.86.0

In der folgenden Tabelle sind die ARNs aufgeführt, die für diese Version der Erweiterung in jeder AWS Region verwendet werden können, in der sie verfügbar ist.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:11`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:11`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:11`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:11`
Asia Pacific (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:11`

Region	ARN
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:11`
Asien-Pazifik (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:11`
Asien-Pazifik (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:11`
Asien-Pazifik (Tokio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:11`
Canada (Central)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:11`
Europe (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:11`
Europa (Irland)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:11`
Europe (London)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:11`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:11`
Europa (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:11`
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:11`

1.0.54.0

Version 1.0.54.0 war die erste Version der Lambda-Insights-Erweiterung.

ARNs für Version 1.0.54.0

In der folgenden Tabelle sind die ARNs aufgeführt, die für diese Version der Erweiterung in jeder AWS Region verwendet werden können, in der sie verfügbar ist.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:2`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:2`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:2`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:2`
Asia Pacific (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:2`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:2`
Asien-Pazifik (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:2`
Asien-Pazifik (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:2`
Asien-Pazifik (Tokio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:2`
Canada (Central)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:2`
Europe (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:2`

Region	ARN
Europa (Irland)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:2`
Europe (London)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:2`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:2`
Europa (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:2`
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:2`

ARM64-Plattformen

In diesem Abschnitt werden die Versionen der Lambda Insights-Erweiterung für ARM64-Plattformen und die ARNs aufgeführt, die für diese Erweiterungen in jeder Region verwendet werden sollen. AWS

> **Important**
>
> Lambda Insights-Erweiterungen 1.0.317.0 und höher unterstützen Amazon Linux 1 nicht.

1.0.317.0

Version 1.0.317.0 beinhaltet die Entfernung der Unterstützung für die Amazon Linux 1-Plattform und Bugfixes. Sie beinhaltet auch Unterstützung für Regionen. AWS GovCloud (US)

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:19`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:21`

Region	ARN
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:17`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:19`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:17`
Asien-Pazifik (Hongkong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:17`
Asien-Pazifik (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension-Arm64:5`
Asien-Pazifik (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:17`
Asien-Pazifik (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:21`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:16`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Asien-Pazifik (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:19`
Asien-Pazifik (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:19`
Asien-Pazifik (Tokio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:30`
Canada (Central)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:17`

Region	ARN
Europe (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:19`
Europa (Irland)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:19`
Europa (London)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:19`
Europa (Milan)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:17`
Europa (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:17`
Europa (Spain)	`arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension-Arm64:5`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:17`
Middle East (Bahrain)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:17`
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:17`
AWS GovCloud (US-Ost)	`arn:aws-us-gov:lambda:us-gov-east-1:122132214140:layer:LambdaInsightsExtension-Arm64:1`
AWS GovCloud (US-West)	`arn:aws-us-gov:lambda:us-gov-west-1:751350123760:layer:LambdaInsightsExtension-Arm64:1`

1.0.295.0

Version 1.0.295.0 enthält Abhängigkeitsupdates für alle kompatiblen Laufzeiten.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:18`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:20`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:16`
Asien-Pazifik (Hongkong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:16`
Asien-Pazifik (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension-Arm64:4`
Asien-Pazifik (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:16`
Asien-Pazifik (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:20`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:15`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:17`
Asien-Pazifik (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:18`

Region	ARN
Asien-Pazifik (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:18
Asien-Pazifik (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:29
Canada (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:16
Europe (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:18
Europa (Irland)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:18
Europa (London)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:18
Europa (Milan)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:16
Europa (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:16
Europa (Spain)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension-Arm64:4
Europe (Stockholm)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:16
Middle East (Bahrain)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:16
Südamerika (São Paulo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:16

1.0.275.0

Version 1.0.275.0 enthält Bugfixes für alle kompatiblen Laufzeiten und unterstützt die Regionen Europa (Spanien) und Asien-Pazifik (Hyderabad).

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:18`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:14`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:16`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:14`
Asien-Pazifik (Hongkong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:14`
Asien-Pazifik (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension-Arm64:2`
Asien-Pazifik (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:14`
Asien-Pazifik (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:13`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:15`

Region	ARN
Asien-Pazifik (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:16
Asien-Pazifik (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:16
Asien-Pazifik (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:27
Canada (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:14
Europe (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:16
Europa (Irland)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:16
Europa (London)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:16
Europa (Milan)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:14
Europa (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:14
Europa (Spain)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension-Arm64:2
Europe (Stockholm)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:14
Middle East (Bahrain)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:14
Südamerika (São Paulo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:14

1.0.273.0

Version 1.0.273.0 enthält Bugfixes für alle kompatiblen Laufzeiten.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:12`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:14`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:9`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:12`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:9`
Asia Pacific (Hongkong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:9`
Asien-Pazifik (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:9`
Asien-Pazifik (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:14`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:9`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:11`
Asien-Pazifik (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:12`

Region	ARN
Asien-Pazifik (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:12
Asien-Pazifik (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:23
Canada (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:10
Europe (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:12
Europa (Irland)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:12
Europa (London)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:12
Europa (Milan)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:9
Europe (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:10
Europe (Stockholm)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:10
Middle East (Bahrain)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:9
Südamerika (São Paulo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:10

1.0.229.0

Version 1.0.229.0 enthält Fehlerbehebungen für alle kompatiblen Laufzeiten. Sie bietet auch Unterstützung für die folgenden Regionen:

- USA West (Nordkalifornien)
- Afrika (Kapstadt)
- Asia Pacific (Hongkong)
- Asien-Pazifik (Jakarta)
- Asien-Pazifik (Osaka)
- Asien-Pazifik (Seoul)
- Kanada (Zentral)
- Europa (Milan)
- Europe (Paris)
- Europe (Stockholm)
- Middle East (Bahrain)
- Südamerika (São Paulo)

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:5`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:7`
USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:3`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Africa (Cape Town)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:2`
Asia Pacific (Hongkong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:2`
Asien-Pazifik (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:2`

Region	ARN
Asien-Pazifik (Mumbai)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:7
Asia Pacific (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:2
Asia Pacific (Seoul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:4
Asien-Pazifik (Singapur)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:5
Asien-Pazifik (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:5
Asien-Pazifik (Tokio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:11
Canada (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:3
Europe (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:5
Europa (Irland)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:5
Europe (London)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:5
Europa (Spain)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:2
Europa (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:3
Europe (Stockholm)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:3

Region	ARN
Middle East (Bahrain)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:2`
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:3`

1.0.135.0

Version 1.0.135.0 enthält Fehlerkorrekturen dafür, wie Lambda Insights die Nutzung von Festplatten- und Dateibeschreibungen sammelt und meldet. In früheren Versionen der Erweiterung meldete die `tmp_free`-Metrik den maximal freien Speicherplatz im `/tmp`-Verzeichnis während der Ausführung einer Funktion. Diese Version ändert die Metrik, um stattdessen den Mindestwert zu melden, was bei der Bewertung der Festplattennutzung nützlicher ist. Weitere Informationen zu Speicherkontingenten für `tmp`-Verzeichnisse finden Sie unter Lambda-Kontingente.

Version 1.0.135.0 meldet jetzt auch die Nutzung von Dateibeschreibungen (`fd_use` und `fd_max`) als prozessübergreifenden Maximalwert, anstatt die Betriebssystemebene zu melden.

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:2`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Asia Pacific (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Asien-Pazifik (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`

Region	ARN
Asien-Pazifik (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Asien-Pazifik (Tokio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Europe (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Europa (Irland)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Europe (London)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:2`

1.0.119.0

Region	ARN
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
USA Ost (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:1`
USA West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Asia Pacific (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Asien-Pazifik (Singapur)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Asien-Pazifik (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:1`

Region	ARN
Asien-Pazifik (Tokio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Europe (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Europa (Irland)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Europe (London)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:1`

Verwenden der Konsole zum Aktivieren von Lambda Insights für eine vorhandene Lambda-Funktion

Führen Sie die folgenden Schritte in der Lambda-Konsole aus, um Lambda Insights für eine vorhandene Lambda-Funktion zu aktivieren.

So aktivieren Sie Lambda Insights für eine Lambda-Funktion

1. Öffnen Sie die Konsole unter https://console.aws.amazon.com/lambda/ AWS Lambda .
2. Wählen Sie den Namen einer Funktion und dann auf dem folgenden Bildschirm die Registerkarte Configuration (Konfiguration) aus.
3. Wählen Sie auf der Registerkarte Konfiguration im linken Navigationsmenü die Option Überwachungs- und Betriebstools und dann Bearbeiten aus.

   Sie werden zu einem Bildschirm weitergeleitet, auf dem Sie Überwachungstools bearbeiten können.
4. Wählen Sie durch Lambda Insights Enhanced Monitoring die Option Bearbeiten aus.
5. Aktivieren Sie unter CloudWatch Lambda Insights die Option Enhanced Monitoring und wählen Sie dann Save aus.

Verwenden von AWS CLI , um Lambda Insights für eine bestehende Lambda-Funktion zu aktivieren

Gehen Sie wie folgt vor, um Lambda Insights für eine bestehende Lambda-Funktion AWS CLI zu aktivieren.

Schritt 1: Aktualisieren der Funktionsberechtigungen

So aktualisieren Sie die Berechtigungen der Funktion

- Fügen Sie die CloudWatchLambdaInsightsExecutionRolePolicyverwaltete IAM-Richtlinie der Ausführungsrolle der Funktion zu, indem Sie den folgenden Befehl eingeben.

```
aws iam attach-role-policy \
--role-name function-execution-role \
--policy-arn "arn:aws:iam::aws:policy/CloudWatchLambdaInsightsExecutionRolePolicy"
```

Schritt 2: Installieren der Lambda-Erweiterung

Installieren Sie die Lambda-Erweiterung, indem Sie den folgenden Befehl eingeben. Ersetzen Sie den ARN-Wert für den `layers`-Parameter durch den ARN, der Ihrer Region und der Erweiterungsversion entspricht, die Sie verwenden möchten. Weitere Informationen finden Sie unter Verfügbare Versionen der Lambda-Insights-Erweiterung.

```
aws lambda update-function-configuration \
   --function-name function-name \
   --layers "arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:14"
```

Schritt 3: Aktivieren Sie den CloudWatch VPC-Endpunkt Logs

Dieser Schritt ist nur für Funktionen erforderlich, die in einem privaten Subnetz ohne Internetzugang ausgeführt werden, und wenn Sie noch keinen CloudWatch Logs Virtual Private Cloud (VPC) - Endpunkt konfiguriert haben.

Wenn Sie diesen Schritt ausführen müssen, geben Sie den folgenden Befehl ein und ersetzen Sie die Platzhalter durch Informationen für Ihre VPC.

Weitere Informationen finden Sie unter Verwenden von CloudWatch Protokollen mit Schnittstellen-VPC-Endpunkten.

```
aws ec2 create-vpc-endpoint \
--vpc-id vpcId \
--vpc-endpoint-type Interface \
--service-name com.amazonaws.region.logs \
--subnet-id subnetId
--security-group-id securitygroupId
```

Verwenden der AWS SAM CLI zur Aktivierung von Lambda Insights für eine bestehende Lambda-Funktion

Gehen Sie wie folgt vor, um Lambda Insights für eine bestehende Lambda-Funktion AWS SAM AWS CLI zu aktivieren.

Wenn Sie die neueste Version der AWS SAM CLI noch nicht installiert haben, müssen Sie sie zuerst installieren oder aktualisieren. Weitere Informationen finden Sie unter Installation der AWS SAM CLI.

Schritt 1: Installieren der Ebene

Um die Lambda-Insights-Erweiterung für alle Ihre Lambda-Funktionen verfügbar zu machen, fügen Sie eine `Layers`-Eigenschaft zum `Globals`-Abschnitt Ihrer SAM-Vorlage mit dem ARN der Lambda-Insights-Ebene hinzu. Im folgenden Beispiel wird die Ebene für die erste Veröffentlichung von Lambda Insights verwendet. Die neueste Release-Version der Lambda-Insights-Erweiterungsebene finden Sie unter Verfügbare Versionen der Lambda-Insights-Erweiterung.

```
Globals:
  Function:
    Layers:
      - !Sub "arn:aws:lambda:
${AWS::Region}:580247275435:layer:LambdaInsightsExtension:14"
```

Um diese Ebene nur für eine einzelne Funktion zu aktivieren, fügen Sie die `Layers`-Eigenschaft wie in diesem Beispiel gezeigt zur Funktion hinzu.

```
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      Layers:
        - !Sub "arn:aws:lambda:
${AWS::Region}:580247275435:layer:LambdaInsightsExtension:14"
```

Schritt 2: Hinzufügen der verwalteten Richtlinie

Fügen Sie für jede Funktion die CloudWatchLambdaInsightsExecutionRolePolicyIAM-Richtlinie hinzu.

AWS SAM unterstützt keine globalen Richtlinien, daher müssen Sie diese für jede Funktion einzeln aktivieren, wie in diesem Beispiel gezeigt. Weitere Informationen zu Globalen finden Sie im Abschnitt Globale.

```
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      Policies:
        - CloudWatchLambdaInsightsExecutionRolePolicy
```

Lokales Aufrufen

Die AWS SAM CLI unterstützt Lambda-Erweiterungen. Jeder lokal ausgeführte Aufruf setzt jedoch die Laufzeitumgebung zurück. Lambda-Insights-Daten sind nicht über lokale Aufrufe verfügbar, da die Laufzeitumgebung ohne ein Herunterfahrungsereignis neu gestartet wird. Weitere Informationen finden Sie unter Version 1.6.0 — Unterstützung für lokales Testen von AWS Lambda Erweiterungen hinzufügen.

Fehlersuche

Um Probleme mit Ihrer Lambda-Insights-Installation zu beheben, fügen Sie Ihrer Lambda-Funktion die folgende Umgebungsvariable hinzu, um die Debug-Protokollierung zu aktivieren.

```
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      Environment:
        Variables:
          LAMBDA_INSIGHTS_LOG_LEVEL: info
```

Wird verwendet AWS CloudFormation , um Lambda Insights für eine bestehende Lambda-Funktion zu aktivieren

Gehen AWS CloudFormation Sie wie folgt vor, um Lambda Insights für eine bestehende Lambda-Funktion zu aktivieren.

Schritt 1: Installieren der Ebene

Fügen Sie die Lambda-Insights-Ebene der `Layers`-Eigenschaft im Lambda-Insights-Ebenen-ARN hinzu. Im folgenden Beispiel wird die Ebene für die erste Veröffentlichung von Lambda Insights verwendet. Die neueste Release-Version der Lambda-Insights-Erweiterungsebene finden Sie unter Verfügbare Versionen der Lambda-Insights-Erweiterung.

```
Resources:
  MyFunction:
    Type: AWS::Lambda::Function
    Properties:
      Layers:
        - !Sub "arn:aws:lambda:
${AWS::Region}:580247275435:layer:LambdaInsightsExtension:14"
```

Schritt 2: Hinzufügen der verwalteten Richtlinie

Fügen Sie die CloudWatchLambdaInsightsExecutionRolePolicyIAM-Richtlinie zu Ihrer Funktionsausführungsrolle hinzu.

```
Resources:
  MyFunctionExecutionRole:
    Type: 'AWS::IAM::Role'
    Properties:
      ManagedPolicyArns:
        - 'arn:aws:iam::aws:policy/CloudWatchLambdaInsightsExecutionRolePolicy'
```

Schritt 3: (Optional) VPC-Endpunkt hinzufügen

Dieser Schritt ist nur für Funktionen erforderlich, die in einem privaten Subnetz ohne Internetzugang ausgeführt werden, und wenn Sie noch keinen CloudWatch Logs Virtual Private Cloud (VPC) - Endpunkt konfiguriert haben. Weitere Informationen finden Sie unter Verwenden von CloudWatch Protokollen mit Schnittstellen-VPC-Endpunkten.

```
Resources:
  CloudWatchLogsVpcPrivateEndpoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      PrivateDnsEnabled: 'true'
      VpcEndpointType: Interface
      VpcId: !Ref: VPC
      ServiceName: !Sub com.amazonaws.${AWS::Region}.logs
```

```
      SecurityGroupIds:
        - !Ref InterfaceVpcEndpointSecurityGroup
      SubnetIds:
        - !Ref PublicSubnet01
        - !Ref PublicSubnet02
        - !Ref PublicSubnet03
```

Verwenden von AWS CDK , um Lambda Insights für eine bestehende Lambda-Funktion zu aktivieren

Gehen Sie wie folgt vor, um Lambda Insights für eine bestehende Lambda-Funktion AWS CDK zu aktivieren. Um diese Schritte verwenden zu können, müssen Sie die bereits AWS CDK zur Verwaltung Ihrer Ressourcen verwenden.

Die Befehle in diesem Abschnitt befinden sich in TypeScript.

Aktualisieren Sie zunächst die Funktionsberechtigungen.

```
executionRole.addManagedPolicy(
 ManagedPolicy.fromAwsManagedPolicyName('CloudWatchLambdaInsightsExecutionRolePolicy')
);
```

Als nächstes installieren Sie die Erweiterung auf der Lambda-Funktion. Ersetzen Sie den ARN-Wert für den `layerArn`-Parameter durch den ARN, der Ihrer Region und der Erweiterungsversion entspricht, die Sie verwenden möchten. Weitere Informationen finden Sie unter Verfügbare Versionen der Lambda-Insights-Erweiterung.

```
import lambda = require('@aws-cdk/aws-lambda');
const layerArn = 'arn:aws:lambda:us-
west-1:580247275435:layer:LambdaInsightsExtension:14';
const layer = lambda.LayerVersion.fromLayerVersionArn(this, 'LayerFromArn', layerArn);
```

Aktivieren Sie bei Bedarf den Virtual Private Cloud (VPC) -Endpunkt für CloudWatch Logs. Dieser Schritt ist nur für Funktionen erforderlich, die in einem privaten Subnetz ohne Internetzugang ausgeführt werden, und wenn Sie noch keinen CloudWatch Logs-VPC-Endpunkt konfiguriert haben.

```
const cloudWatchLogsEndpoint = vpc.addInterfaceEndpoint('cwl-gateway', {
  service: InterfaceVpcEndpointAwsService.CLOUDWATCH_LOGS,
});

cloudWatchLogsEndpoint.connections.allowDefaultPortFromAnyIpv4();
```

Verwenden des Serverless-Frameworks zum Aktivieren von Lambda Insights für eine vorhandene Lambda-Funktion

Führen Sie diese Schritte aus, um mit Serverless Framework Lambda Insights für eine vorhandene Lambda-Funktion zu aktivieren. Weitere Informationen zu Serverless Framework finden Sie unter serverless.com.

Dies geschieht über ein Lambda-Insights-Plugin für Serverless. Weitere Informationen finden Sie unter. serverless-plugin-lambda-insights

Wenn Sie die neueste Version der Serverless-Befehlszeilenschnittstelle noch nicht installiert haben, müssen Sie sie zuerst installieren oder aktualisieren. Weitere Informationen finden Sie unter Erste Schritte mit Serverless Framework Open Source & AWS.

So verwenden Sie Serverless Framework zum Aktivieren von Lambda Insights für eine Lambda-Funktion

1. Installieren Sie das Serverless-Plug-In für Lambda Insights, indem Sie den folgenden Befehl in Ihrem Serverless-Verzeichnis ausführen:

   ```
   npm install --save-dev serverless-plugin-lambda-insights
   ```

2. Fügen Sie in Ihrer Datei `serverless.yml` das Plug-In im Abschnitt `plugins` wie gezeigt hinzu:

   ```
   provider:
     name: aws
   plugins:
     - serverless-plugin-lambda-insights
   ```

3. Aktivieren Sie Lambda Insights.

   - Sie können Lambda Insights einzeln pro Funktion aktivieren, indem Sie der Datei serverless.yml die folgende Eigenschaft hinzufügen

     ```
     functions:
       myLambdaFunction:
         handler: src/app/index.handler
         lambdaInsights: true #enables Lambda Insights for this function
     ```

- Sie können Lambda Insights für alle Funktionen in der Datei `serverless.yml` aktivieren, indem Sie den folgenden benutzerdefinierten Abschnitt hinzufügen:

```
custom:
  lambdaInsights:
    defaultLambdaInsights: true #enables Lambda Insights for all functions
```

4. Stellen Sie den Serverless-Service erneut bereit, indem Sie den folgenden Befehl eingeben:

```
serverless deploy
```

Dadurch werden alle Funktionen neu bereitgestellt und Lambda Insights für die von Ihnen angegebenen Funktionen aktiviert. Es aktiviert Lambda Insights, indem es die Lambda-Insights-Ebene hinzufügt und die erforderlichen Berechtigungen mithilfe der `arn:aws:iam::aws:policy/CloudWatchLambdaInsightsExecutionRolePolicy` IAM-Richtlinie anfügt.

Aktivieren von Lambda Insights für eine Lambda-Container-Image-Bereitstellung

Um Lambda Insights für eine Lambda-Funktion zu aktivieren, die als Container-Image bereitgestellt wird, fügen Sie Zeilen in Ihre Dockerdatei ein. Mit diesen Zeilen wird der Lambda-Insights-Agent als Erweiterung in Ihrem Container-Image installiert. Die hinzuzufügenden Zeilen unterscheiden sich für x86-64-Container und ARM64-Container.

> **Note**
>
> Der Lambda-Insights-Agent wird nur bei Lambda-Laufzeiten unterstützt, die Amazon Linux 2 verwenden.

Themen

Bereitstellung eines x86-64-Container-Images

Um Lambda Insights für eine Lambda-Funktion zu aktivieren, die als Container-Image, das auf einem x86-64-Container läuft, bereitgestellt wird, fügen Sie die folgenden Zeilen in Ihre Dockerdatei ein. Mit diesen Zeilen wird der Lambda-Insights-Agent als Erweiterung in Ihrem Container-Image installiert.

```
RUN curl -O https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension.rpm && \
    rpm -U lambda-insights-extension.rpm && \
    rm -f lambda-insights-extension.rpm
```

Nachdem Sie Ihre Lambda-Funktion erstellt haben, weisen Sie die CloudWatchLambdaInsightsExecutionRolePolicyIAM-Richtlinie der Ausführungsrolle der Funktion zu, und Lambda Insights wird für die auf Container-Images basierende Lambda-Funktion aktiviert.

> **Note**
>
> Um eine ältere Version der Lambda Insights-Erweiterung zu verwenden, ersetzen Sie die URL in den vorherigen Befehlen durch diese URL: `https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension.1.0.111.0.rpm`. Derzeit sind nur Lambda Insights Versionen 1.0.111.0 und höher verfügbar. Weitere Informationen finden Sie unter Verfügbare Versionen der Lambda-Insights-Erweiterung.

So überprüfen Sie die Signatur des Lambda-Insights-Agent-Pakets auf einem Linux-Server

1. Geben Sie den folgenden Befehl ein, um den öffentlichen Schlüssel herunterzuladen.

   ```
   shell$ wget https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
   lambda-insights-extension.gpg
   ```

2. Geben Sie den folgenden Befehl ein, um den öffentlichen Schlüssel in Ihren Schlüsselbund zu importieren.

   ```
   shell$ gpg --import lambda-insights-extension.gpg
   ```

   Die Ausgabe sieht folgendermaßen oder ähnlich aus: Notieren Sie sich den `key`-Wert, den Sie im nächsten Schritt benötigen. In dieser Beispielausgabe ist der Schlüssel-Wert `848ABDC8`.

```
gpg: key 848ABDC8: public key "Amazon Lambda Insights Extension" imported
gpg: Total number processed: 1
gpg: imported: 1  (RSA: 1)
```

3. Überprüfen Sie den Fingerabdruck, indem Sie den folgenden Befehl eingeben. Ersetzen Sie `key-value` durch den Wert des Schlüssels aus dem vorherigen Schritt.

```
shell$  gpg --fingerprint key-value
```

   Die Fingerabdruck-Zeichenfolge in der Ausgabe dieses Befehls sollte `E0AF FA11 FFF3 5BD7 349E E222 479C 97A1 848A BDC8` sein. Wenn die Zeichenfolge nicht übereinstimmt, installieren Sie den Agenten nicht und wenden Sie sich an AWS.

4. Nachdem Sie den Fingerabdruck verifiziert haben, können Sie ihn verwenden, um das Lambda-Insights-Agenten-Paket zu verifizieren. Laden Sie die Paket-Signaturdatei herunter, indem Sie den folgenden Befehl eingeben.

```
shell$  wget https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension.rpm.sig
```

5. Überprüfen Sie die Signatur, indem Sie den folgenden Befehl eingeben.

```
shell$ gpg --verify lambda-insights-extension.rpm.sig lambda-insights-extension.rpm
```

   Die Ausgabe sollte wie folgt aussehen:

```
gpg: Signature made Thu 08 Apr 2021 06:41:00 PM UTC using RSA key ID 848ABDC8
gpg: Good signature from "Amazon Lambda Insights Extension"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: E0AF FA11 FFF3 5BD7 349E  E222 479C 97A1 848A BDC8
```

   In der erwarteten Ausgabe wird möglicherweise eine Warnung über eine vertrauenswürdige Signatur angezeigt. Beachten Sie die Warnung zu vertrauenswürdigen Inhalten. Das bedeutet nicht, dass die Signatur ungültig ist, sondern nur, dass Sie den öffentlichen Schlüssel nicht überprüft haben.

Wenn die Ausgabe `BAD signature` enthält, überprüfen Sie, ob Sie die Schritte richtig ausgeführt haben. Wenn Sie weiterhin eine `BAD signature` Antwort erhalten, kontaktieren Sie uns AWS und vermeiden Sie es, die heruntergeladene Datei zu verwenden.

x86-64-Beispiel

Dieser Abschnitt enthält ein Beispiel für die Aktivierung von Lambda Insights für eine auf Container-Images basierende Python-Lambda-Funktion.

Ein Beispiel für die Aktivierung von Lambda Insights für ein Lambda-Container-Image

1. Erstellen Sie eine Dockerdatei, die der folgenden ähnlich ist:

```
FROM public.ecr.aws/lambda/python:3.8

// extra lines to install the agent here
RUN curl -O https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension.rpm && \
    rpm -U lambda-insights-extension.rpm && \
    rm -f lambda-insights-extension.rpm

COPY index.py ${LAMBDA_TASK_ROOT}
CMD [ "index.handler" ]
```

2. Erstellen Sie eine Python-Datei namens `index.py`, die der folgenden ähnelt:

```
def handler(event, context):
  return {
    'message': 'Hello World!'
  }
```

3. Legen Sie Dockerfile und `index.py` in dasselbe Verzeichnis. Führen Sie dann in diesem Verzeichnis die folgenden Schritte aus, um das Docker-Image zu erstellen und es in Amazon ECR hochzuladen.

```
// create an ECR repository
aws ecr create-repository --repository-name test-repository
// build the docker image
docker build -t test-image .
// sign in to AWS
```

```
aws ecr get-login-password | docker login --username AWS --password-stdin
 "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com
// tag the image
docker tag test-image:latest "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/
test-repository:latest
// push the image to ECR
docker push "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/test-
repository:latest
```

4. Verwenden Sie das Amazon-ECR-Image, das Sie gerade erstellt haben, um die Lambda-Funktion zu erstellen.
5. Weisen Sie die CloudWatchLambdaInsightsExecutionRolePolicyIAM-Richtlinie der Ausführungsrolle der Funktion zu.

ARM64-Container-Image-Bereitstellung

Um Lambda Insights für eine Lambda-Funktion zu aktivieren, die als Container-Image, das auf einem AL2_aarch64-Container läuft (der die ARM64-Architektur verwendet), bereitgestellt wird, fügen Sie die folgenden Zeilen in Ihre Dockerdatei ein. Mit diesen Zeilen wird der Lambda-Insights-Agent als Erweiterung in Ihrem Container-Image installiert.

```
RUN curl -O https://lambda-insights-extension-arm64.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension-arm64.rpm && \
    rpm -U lambda-insights-extension-arm64.rpm && \
    rm -f lambda-insights-extension-arm64.rpm
```

Nachdem Sie Ihre Lambda-Funktion erstellt haben, weisen Sie die CloudWatchLambdaInsightsExecutionRolePolicyIAM-Richtlinie der Ausführungsrolle der Funktion zu, und Lambda Insights wird für die auf Container-Images basierende Lambda-Funktion aktiviert.

> **Note**
>
> Um eine ältere Version der Lambda Insights-Erweiterung zu verwenden, ersetzen Sie die URL in den vorherigen Befehlen durch diese URL: `https://lambda-insights-extension-arm64.s3-ap-northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension-arm64.1.0.229.0.rpm`. Derzeit sind nur Lambda-Insights-Versionen 1.0.229.0 und höher verfügbar. Weitere Informationen finden Sie unter Verfügbare Versionen der Lambda-Insights-Erweiterung.

So überprüfen Sie die Signatur des Lambda-Insights-Agent-Pakets auf einem Linux-Server

1. Geben Sie den folgenden Befehl ein, um den öffentlichen Schlüssel herunterzuladen.

```
shell$ wget https://lambda-insights-extension-arm64.s3-ap-
northeast-1.amazonaws.com/lambda-insights-extension.gpg
```

2. Geben Sie den folgenden Befehl ein, um den öffentlichen Schlüssel in Ihren Schlüsselbund zu importieren.

```
shell$ gpg --import lambda-insights-extension.gpg
```

   Die Ausgabe sieht folgendermaßen oder ähnlich aus: Notieren Sie sich den `key`-Wert, den Sie im nächsten Schritt benötigen. In dieser Beispielausgabe ist der Schlüssel-Wert `848ABDC8`.

```
gpg: key 848ABDC8: public key "Amazon Lambda Insights Extension" imported
gpg: Total number processed: 1
gpg: imported: 1  (RSA: 1)
```

3. Überprüfen Sie den Fingerabdruck, indem Sie den folgenden Befehl eingeben. Ersetzen Sie `key-value` durch den Wert des Schlüssels aus dem vorherigen Schritt.

```
shell$  gpg --fingerprint key-value
```

   Die Fingerabdruck-Zeichenfolge in der Ausgabe dieses Befehls sollte `E0AF FA11 FFF3 5BD7 349E E222 479C 97A1 848A BDC8` sein. Wenn die Zeichenfolge nicht übereinstimmt, installieren Sie den Agenten nicht und wenden Sie sich an AWS.

4. Nachdem Sie den Fingerabdruck verifiziert haben, können Sie ihn verwenden, um das Lambda-Insights-Agenten-Paket zu verifizieren. Laden Sie die Paket-Signaturdatei herunter, indem Sie den folgenden Befehl eingeben.

```
shell$  wget https://lambda-insights-extension-arm64.s3-ap-
northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension-arm64.rpm.sig
```

5. Überprüfen Sie die Signatur, indem Sie den folgenden Befehl eingeben.

```
shell$ gpg --verify lambda-insights-extension-arm64.rpm.sig lambda-insights-
extension-arm64.rpm
```

Die Ausgabe sollte wie folgt aussehen:

```
gpg: Signature made Thu 08 Apr 2021 06:41:00 PM UTC using RSA key ID 848ABDC8
gpg: Good signature from "Amazon Lambda Insights Extension"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: E0AF FA11 FFF3 5BD7 349E  E222 479C 97A1 848A BDC8
```

In der erwarteten Ausgabe wird möglicherweise eine Warnung über eine vertrauenswürdige Signatur angezeigt. Beachten Sie die Warnung zu vertrauenswürdigen Inhalten. Das bedeutet nicht, dass die Signatur ungültig ist, sondern nur, dass Sie den öffentlichen Schlüssel nicht überprüft haben.

Wenn die Ausgabe `BAD signature` enthält, überprüfen Sie, ob Sie die Schritte richtig ausgeführt haben. Wenn Sie weiterhin eine `BAD signature` Antwort erhalten, kontaktieren Sie uns AWS und vermeiden Sie es, die heruntergeladene Datei zu verwenden.

ARM64-Beispiel

Dieser Abschnitt enthält ein Beispiel für die Aktivierung von Lambda Insights für eine auf Container-Images basierende Python-Lambda-Funktion.

Ein Beispiel für die Aktivierung von Lambda Insights für ein Lambda-Container-Image

1. Erstellen Sie eine Dockerdatei, die der folgenden ähnlich ist:

```
FROM public.ecr.aws/lambda/python:3.8
// extra lines to install the agent here
RUN curl -O https://lambda-insights-extension-arm64.s3-ap-
northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension-arm64.rpm && \
    rpm -U lambda-insights-extension-arm64.rpm && \
    rm -f lambda-insights-extension-arm64.rpm

COPY index.py ${LAMBDA_TASK_ROOT}
CMD [ "index.handler" ]
```

2. Erstellen Sie eine Python-Datei namens `index.py`, die der folgenden ähnelt:

```
def handler(event, context):
  return {
```

```
    'message': 'Hello World!'
  }
```

3. Legen Sie Dockerfile und `index.py` in dasselbe Verzeichnis. Führen Sie dann in diesem Verzeichnis die folgenden Schritte aus, um das Docker-Image zu erstellen und es in Amazon ECR hochzuladen.

```
// create an ECR repository
aws ecr create-repository --repository-name test-repository
// build the docker image
docker build -t test-image .
// sign in to AWS
aws ecr get-login-password | docker login --username AWS --password-stdin
 "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com
// tag the image
docker tag test-image:latest "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/
test-repository:latest
// push the image to ECR
docker push "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/test-
repository:latest
```

4. Verwenden Sie das Amazon-ECR-Image, das Sie gerade erstellt haben, um die Lambda-Funktion zu erstellen.
5. Weisen Sie die CloudWatchLambdaInsightsExecutionRolePolicyIAM-Richtlinie der Ausführungsrolle der Funktion zu.

Anzeigen Ihrer Lambda-Insights-Metriken

Nachdem Sie die Lambda Insights-Erweiterung auf einer aufgerufenen Lambda-Funktion installiert haben, können Sie die CloudWatch Konsole verwenden, um Ihre Metriken anzuzeigen. Sie können eine Multifunktionsübersicht sehen oder sich auf eine einzelne Funktion konzentrieren.

Eine Liste der Lambda-Insights-Metriken finden Sie unter Von Lambda Insights erfasste Metriken.

So zeigen Sie die Multifunktionsübersicht Ihrer Lambda-Insights-Metriken an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Klicken Sie im linken Navigationsbereich unter Lambda Insights auf Multifunktions-Funktion.

Im oberen Teil der Seite werden Diagramme mit aggregierten Metriken aller Lambda-Funktionen in der Region angezeigt, für die Lambda Insights aktiviert sind. Unten auf der Seite ist eine Tabelle, die die Funktionen auflistet.

3. Um nach Funktionsnamen zu filtern, um die Anzahl der angezeigten Funktionen zu reduzieren, geben Sie einen Teil des Funktionsnamens in das Feld am oberen Rand der Seite ein.
4. Um diese Ansicht einem Dashboard als Widget hinzuzufügen, wählen Sie Hinzufügen zu Dashboard aus.

So zeigen Sie Metriken für eine einzelne Funktion an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Klicken Sie im linken Navigationsbereich unter Lambda Insights auf Einzelne Funktion.

   Im oberen Teil der Seite werden Diagramme mit Metriken für die ausgewählte Funktion angezeigt.
3. Wenn X-Ray aktiviert ist, können Sie eine einzelne Ablaufverfolgungs-ID auswählen. Dadurch wird die Seite der X-Ray-Trace-Karte für diesen Aufruf geöffnet und von dort aus können Sie herauszoomen, um die verteilte Ablaufverfolgung und die anderen Services anzuzeigen, die an der Verarbeitung dieser bestimmten Transaktion beteiligt sind. Weitere Informationen zur X-Ray-Trace-Karte finden Sie unter Verwenden der X-Ray-Trace-Karte.
4. Um CloudWatch Logs Insights zu öffnen und einen bestimmten Fehler zu vergrößern, wählen Sie in der Tabelle unten auf der Seite die Option Logs anzeigen aus.
5. Um diese Ansicht einem Dashboard als Widget hinzuzufügen, wählen Sie Hinzufügen zu Dashboard aus.

Integration in Application Insights

Amazon CloudWatch Application Insights unterstützt Sie bei der Überwachung Ihrer Anwendungen und identifiziert und richtet wichtige Kennzahlen, Protokolle und Alarme für Ihre Anwendungsressourcen und Ihren Technologie-Stack ein. Weitere Informationen finden Sie unter Einblicke in CloudWatch Amazon-Anwendungen.

Sie können Application Insights aktivieren, um zusätzliche Daten aus Ihren Lambda-Funktionen zu sammeln. Sofern noch nicht erfolgt, können Sie es unter der Leistungsansicht auf dem Lambda-Insights-Dashboard aktivieren, indem Sie Auto-configure Application Insights (Application Insights

automatisch konfigurieren) auf der Registerkarte Application Insights (Application Insights) auswählen.

Wenn Sie CloudWatch Application Insights bereits für die Überwachung Ihrer Lambda-Funktionen eingerichtet haben, wird das Application Insights-Dashboard unter dem Lambda Insights-Dashboard auf der Registerkarte Application Insights angezeigt.

Von Lambda Insights erfasste Metriken

Lambda Insights sammelt mehrere Metriken aus den Lambda-Funktionen, in denen es installiert ist. Einige dieser Metriken sind als aggregierte Zeitreihendaten unter Metriken verfügbar. CloudWatch Andere Metriken werden nicht zu Zeitreihendaten zusammengefasst, können aber mithilfe CloudWatch von Logs Insights in den Logeinträgen im eingebetteten Metrikformat gefunden werden.

Die folgenden Metriken sind als aggregierte Zeitreihendaten unter CloudWatch Metriken im `LambdaInsights` Namespace verfügbar.

Metrikname	Dimensionen	Beschreibung
`cpu_total_time`	function_name function_name, Version	Summe von `cpu_system_time` und `cpu_user_time` . Einheit: Millisekunden
`init_duration`	function_name function_name, Version	Die in der Phase `init` des Lebenszyklus der Lambda-Ausführungsumgebung verbrachte Zeit. Einheit: Millisekunden
`memory_utilization`	function_name function_name, Version	Der maximale Speicher, gemessen als Prozentsatz des der Funktion

Metrikname	Dimensionen	Beschreibung
		zugewiesenen Speicherbereichs. Einheit: Prozent
`rx_bytes`	function_name function_name, Version	Die Anzahl der von der Funktion empfangenen Byte Einheit: Byte
`tmp_used`		Der im Verzeichnis `/tmp` genutzte Speicherplatz. Einheit: Byte
`tx_bytes`	function_name function_name, Version	Die Anzahl der von der Funktion gesendeten Bytes. Einheit: Byte
`total_memory`	function_name function_name, Version	Der Speicherplatz in MB, der Ihrer Lambda-Funktion zugewiesen wird. Dies ist die gleiche wie die Speichergröße Ihrer Funktion. Einheit: Megabyte

Metrikname	Dimensionen	Beschreibung
`total_network`	function_name function_name, Version	Summe von `rx_bytes` und `tx_bytes`. Selbst für Funktionen, die keine I/O-Aufgaben ausführen, ist dieser Wert normalerweise größer als Null, da Netzwerkaufrufe von der Lambda-Laufzeitumgebung ausgeführt werden. Einheit: Byte
`used_memory_max`	function_name function_name, Version	Der gemessene Speicher der Funktions-Sandbox. Einheit: Megabyte

Die folgenden Metriken können mithilfe CloudWatch von Logs Insights in den Protokolleinträgen im eingebetteten Metrikformat gefunden werden. Weitere Informationen zu CloudWatch Logs Insights finden Sie unter Analysieren von Protokolldaten mit CloudWatch Logs Insights.

Weitere Hinweise zum eingebetteten Metrik-Format finden Sie unter Einbetten von Metriken in Protokollen.

Metrikname	Beschreibung	
`cpu_system_time`	Die Zeitspanne, die die CPU für die Ausführung des Kernelcodes verbrachte. Einheit: Millisekunden	

Metrikname	Beschreibung
`cpu_total_time`	Summe von `cpu_system_time` und `cpu_user_time` . Einheit: Millisekunden
`cpu_user_time`	Die Zeitspanne, die die CPU für die Ausführung des Benutzercodes verbrachte. Einheit: Millisekunden
`fd_max`	Die maximale Anzahl der verfügbaren Dateideskriptoren. Einheit: Anzahl
`fd_use`	Die maximale Anzahl der verwendeten Dateideskriptoren. Einheit: Anzahl
`memory_utilization`	Der maximale Speicher, gemessen als Prozentsatz des der Funktion zugewiesenen Speicherbereichs. Einheit: Prozent
`rx_bytes`	Die Anzahl der von der Funktions empfangenen Byte Einheit: Byte
`tx_bytes`	Die Anzahl der von der Funktion gesendeten Bytes. Einheit: Byte

Metrikname	Beschreibung
`threads_max`	Die Anzahl der Threads, die vom Funktions prozess verwendet werden. Als Funktionsautor steuern Sie nicht die anfängliche Anzahl von Threads, die von der Laufzeit erstellt wurden. Einheit: Anzahl
`tmp_max`	Der im Verzeichnis `/tmp` verfügbare Speicherp latz. Einheit: Byte
`total_memory`	Der Speicherplatz in MB, der Ihrer Lambda-Fu nktion zugewiesen wird. Dies ist die gleiche wie die Speichergröße Ihrer Funktion. Einheit: Megabyte
`total_network`	Summe von `rx_bytes` und `tx_bytes`. Selbst für Funktionen, die keine I/O-Aufga ben ausführen, ist dieser Wert normalerweise größer als Null, da Netzwerkaufrufe von der Lambda-Laufzeitumgebung ausgeführt werden. Einheit: Byte
`used_memory_max`	Der gemessene Speicher der Funktions-Sandbox. Einheit: Byte

Problembehandlung und bekannte Probleme

Der erste Schritt zur Behebung von Problemen besteht darin, die Debug-Protokollierung für die Lambda-Insights-Erweiterung zu aktivieren. Legen Sie dazu die folgende Umgebungsvariable für Ihre Lambda Funktion fest: `LAMBDA_INSIGHTS_LOG_LEVEL=info`. Weitere Informationen finden Sie unter Verwenden von AWS Lambda -Umgebungsvariablen.

Die Erweiterung gibt Protokolle in dieselbe Protokollgruppe wie Ihre Funktion aus (`/aws/lambda/function-name`) Überprüfen Sie diese Protokolle, um festzustellen, ob der Fehler möglicherweise mit einem Setup-Problem zusammenhängt.

Ich sehe keine Metriken von Lambda Insights

Wenn die erwarteten Lambda-Insights-Metriken nicht angezeigt werden, prüfen Sie die folgenden Möglichkeiten:

- Die Metriken sind möglicherweise nur verzögert — Wenn die Funktion noch nicht aufgerufen wurde oder die Daten noch nicht gelöscht wurden, werden die Metriken nicht angezeigt. CloudWatch Weitere Informationen finden Sie unter Bekannte Probleme unten in diesem Abschnitt.
- Stellen Sie sicher, dass die Lambda-Funktion über die richtigen Berechtigungen verfügt — Stellen Sie sicher, dass die CloudWatchLambdaInsightsExecutionRolePolicyIAM-Richtlinie der Ausführungsrolle der Funktion zugewiesen ist.
- Überprüfen der Lambda-Laufzeit – Lambda Insights unterstützt nur bestimmte Lambda-Laufzeiten. Eine Liste der unterstützten Laufzeiten finden Sie unter Lambda Insights.

  Um beispielsweise Lambda Insights auf Java 8 zu verwenden, müssen Sie die `java8.al2`-Laufzeit verwenden, nicht die `java8`-Laufzeit.
- Netzwerkzugriff prüfen — Die Lambda-Funktion befindet sich möglicherweise in einem privaten VPC-Subnetz ohne Internetzugang und Sie haben keinen VPC-Endpunkt für Logs konfiguriert. CloudWatch Um dieses Problem zu beheben, können Sie die Umgebungsvariable `LAMBDA_INSIGHTS_LOG_LEVEL=info` festlegen.

Bekannte Probleme

Die Datenverzögerung kann bis zu 20 Minuten betragen. Wenn ein Funktionshandler abgeschlossen ist, friert Lambda die Sandbox ein, wodurch auch die Lambda-Insights-Erweiterung fixiert wird. Während die Funktion läuft, verwenden wir eine adaptive Batching-Strategie basierend auf der Funktion TPS, um Daten auszugeben. Wenn die Funktion jedoch für einen längeren Zeitraum nicht mehr aufgerufen wird und sich noch Ereignisdaten im Puffer befinden, können diese Daten verzögert werden, bis Lambda die Leerlauf-Sandbox herunterfährt. Wenn Lambda die Sandbox herunterfährt, leeren wir die gepufferten Daten.

Beispiel-Telemetrieereignis

Jeder Aufruf einer Lambda-Funktion, für die Lambda Insights aktiviert ist, schreibt ein einzelnes Protokollereignis in die Protokollgruppe `/aws/lambda-insights`. Jedes Protokollereignis enthält Metriken im eingebetteten Metrikformat. Weitere Hinweise zum eingebetteten Metrik-Format finden Sie unter Einbetten von Metriken in Protokollen.

Um diese Protokollereignisse zu analysieren, können Sie die folgenden Methoden verwenden:

- Der Lambda Insights-Bereich der CloudWatch Konsole, wie unter erklärt. Anzeigen Ihrer Lambda-Insights-Metriken
- Protokollieren Sie Ereignisabfragen mit CloudWatch Logs Insights. Weitere Informationen finden Sie unter Analysieren von Protokolldaten mit CloudWatch Logs Insights.
- Im `LambdaInsights` Namespace gesammelte Metriken, die Sie mithilfe von CloudWatch Metriken grafisch darstellen.

Im Folgenden finden Sie ein Beispiel für ein Lambda-Insights-Protokollereignis mit eingebettetem Metriken.

```
{
    "_aws": {
        "Timestamp": 1605034324256,
        "CloudWatchMetrics": [
            {
                "Namespace": "LambdaInsights",
                "Dimensions": [
                    [ "function_name" ],
                    [ "function_name", "version" ]
                ],
                "Metrics": [
                    { "Name": "memory_utilization", "Unit": "Percent" },
                    { "Name": "total_memory", "Unit": "Megabytes" },
                    { "Name": "used_memory_max", "Unit": "Megabytes" },
                    { "Name": "cpu_total_time", "Unit": "Milliseconds" },
                    { "Name": "tx_bytes", "Unit": "Bytes" },
                    { "Name": "rx_bytes", "Unit": "Bytes" },
                    { "Name": "total_network", "Unit": "Bytes" },
                    { "Name": "init_duration", "Unit": "Milliseconds" }
                ]
            }
```

```
        ],
        "LambdaInsights": {
            "ShareTelemetry": true
        }
    },
    "event_type": "performance",
    "function_name": "cpu-intensive",
    "version": "Blue",
    "request_id": "12345678-8bcc-42f7-b1de-123456789012",
    "trace_id": "1-5faae118-12345678901234567890",
    "duration": 45191,
    "billed_duration": 45200,
    "billed_mb_ms": 11571200,
    "cold_start": true,
    "init_duration": 130,
    "tmp_free": 538329088,
    "tmp_max": 551346176,
    "threads_max": 11,
    "used_memory_max": 63,
    "total_memory": 256,
    "memory_utilization": 24,
    "cpu_user_time": 6640,
    "cpu_system_time": 50,
    "cpu_total_time": 6690,
    "fd_use": 416,
    "fd_max": 32642,
    "tx_bytes": 4434,
    "rx_bytes": 6911,
    "timeout": true,
    "shutdown_reason": "Timeout",
    "total_network": 11345,
    "agent_version": "1.0.72.0",
    "agent_memory_avg": 10,
    "agent_memory_max": 10
}
```

Verwenden Sie Contributor Insights, um Daten mit hoher Kardinalität zu analysieren

Mit Contributor Insights können Sie Protokolldaten analysieren und Zeitreihen erstellen, die Daten der Contributors anzeigen. Sie können Metriken über die Top-N-Contributors, die Gesamtzahl der eindeutigen Contributors und deren Nutzung anzeigen. Auf diese Weise können Sie Top-Benutzer

finden und verstehen, wer oder was sich auf die Systemleistung auswirkt. Beispielsweise können Sie fehlerhafte Hosts finden, die stärksten Netzwerkbenutzer identifizieren oder URLs finden, die die meisten Fehler verursachen.

Sie können Ihre Regeln von Grund auf neu erstellen, und wenn Sie die verwenden, können AWS Management Console Sie auch selbst erstellte Beispielregeln verwenden. AWS Regeln definieren die Protokollfelder, die Sie zum Definieren von Contributors verwenden möchten, z. B. `IpAddress`. Sie können die Protokolldaten auch filtern, um das Verhalten einzelner Contributors zu finden und zu analysieren.

CloudWatch bietet auch integrierte Regeln, mit denen Sie Metriken aus anderen AWS Diensten analysieren können.

Alle Regeln analysieren eingehende Daten in Echtzeit.

Wenn Sie mit einem Konto angemeldet sind, das als Überwachungskonto für CloudWatch kontoübergreifende Observability eingerichtet ist, können Sie in diesem Überwachungskonto Contributor Insights-Regeln erstellen, die Protokollgruppen in Quellkonten und im Überwachungskonto analysieren. Sie können auch eine einzelne Regel erstellen, die Protokollgruppen in mehreren Konten analysiert. Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.

> **Note**
>
> Wenn Sie Contributor Insights verwenden, wird Ihnen jedes Auftreten eines Protokollereignisses berechnet, das einer Regel entspricht. Weitere Informationen finden Sie unter CloudWatch Amazon-Preise.

Themen

Eine Contributor-Insights-Regel erstellen

Sie können Regeln zum Analysieren von Protokolldaten erstellen. Alle Protokolle im JSON- oder CLF-Format (Common Log Format) können ausgewertet werden. Dazu gehören Ihre benutzerdefinierten Protokolle, die einem dieser Formate folgen, und Protokolle von AWS Diensten wie Amazon VPC-Flussprotokollen, Amazon Route 53-DNS-Abfrageprotokollen, Amazon ECS-Container-Protokollen und Protokollen von Amazon AWS CloudTrail SageMaker, Amazon RDS AWS AppSync und API Gateway.

Wenn Sie in einer Regel Feldnamen oder -werte angeben, wird bei allen Übereinstimmungen die Groß- und Kleinschreibung beachtet.

Sie können integrierte Beispielregeln verwenden, wenn Sie eine Regel erstellen, oder Ihre eigene Regel von Grund auf neu erstellen. Contributor Insights enthält Beispielregeln für die folgenden Protokolltypen:

- Amazon-API-Gateway-Protokolle
- Öffentliche DNS-Abfrageprotokolle von Amazon Route 53
- Amazon-Route-53-Resolver-Abfrageprotokolle
- CloudWatch Container Insights-Protokolle
- VPC Flow Logs

Wenn Sie mit einem Konto angemeldet sind, das als Überwachungskonto mit CloudWatch kontenübergreifender Observability eingerichtet ist, können Sie Contributor Insights-Regeln für Protokollgruppen in den Quellkonten erstellen, die mit diesem Überwachungskonto verknüpft sind, zusätzlich zu den Regeln für Protokollgruppen im Überwachungskonto. Sie können auch eine einzelne Regel einrichten, die Protokollgruppen in verschiedenen Konten überwacht. Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.

> **Important**
>
> Wenn Sie einem Benutzer die `cloudwatch:PutInsightRule` Berechtigung erteilen, kann dieser Benutzer standardmäßig eine Regel erstellen, die jede Protokollgruppe in Logs auswertet. CloudWatch Sie können IAM-Richtlinienbedingungen hinzufügen, die diese Berechtigungen für einen Benutzer einschränken, um bestimmte Protokollgruppen einzuschließen und auszuschließen. Weitere Informationen finden Sie unter Verwenden

von Bedingungsschlüsseln, um den Zugriff von Contributor-Insights-Benutzern auf Protokollgruppen einzuschränken.

So erstellen Sie eine Regel mit einer integrierten Beispielregel:

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Instances und anschließend Contributor Insights aus.
3. Wählen Sie Regel erstellen aus.
4. Wählen Sie für Log group(s) (Protokollgruppe(n)) die Protokollgruppe(n) aus, die von der Regel überwacht werden sollen. Sie können bis zu 20 Protokollgruppen auswählen. Wenn Sie mit einem Überwachungskonto angemeldet sind, das für CloudWatch kontenübergreifende Beobachtung eingerichtet ist, können Sie Protokollgruppen in Quellkonten auswählen und auch eine einzige Regel einrichten, um Protokollgruppen in verschiedenen Konten zu analysieren.
   - (Optional) Um alle Protokollgruppen auszuwählen, die Namen haben, die mit einer bestimmten Zeichenfolge beginnen, wählen Sie aus dem Dropdown-Menü Select by prefix match (Nach Präfixübereinstimmung auswählen) aus und geben Sie dann das Präfix ein. Wenn es sich um ein Überwachungskonto handelt, können Sie optional die Konten auswählen, in denen gesucht werden soll. Andernfalls werden alle Konten ausgewählt.

   > **Note**
   >
   > Für jedes Protokollereignis, das Ihrer Regel entspricht, fallen Gebühren an. Wenn Sie aus dem Dropdown-Menü Select by prefix match (Nach Präfixübereinstimmung auswählen) auswählen, sollten Sie sich bewusst sein, mit wie vielen Protokollgruppen das Präfix übereinstimmen kann. Wenn Sie mehr Protokollgruppen suchen, als Sie beabsichtigten, können unerwartete Gebühren anfallen. Weitere Informationen finden Sie unter CloudWatch Amazon-Preise.
5. Wählen Sie als Rule type (Regeltyp) die Option Sample rule (Beispielregel) aus. Wählen Sie dann Select sample rule (Beispielregel auswählen) und wählen Sie die Regel aus.
6. Die Beispielregel hat die Log format (Protokollformat), Contribution (Beitrag), Filters (Filter) und Aggregate on (Aggregieren auf) ausgefüllt. Sie können diese Werte anpassen, wenn Sie möchten.

7. Wählen Sie Weiter aus.
8. Geben Sie unter Rule name (Name der Regel) einen Namen ein. Gültige Zeichen sind A-Z, a-z, 0-9, (Bindestrich), (Unterstrich) und (Punkt).
9. Wählen Sie aus, ob die Regel in einem deaktivierten oder aktivierten Zustand erstellt werden soll. Wenn Sie sie aktivieren, beginnt sie sofort mit der Analyse Ihrer Daten. Wenn Sie aktivierte Regeln ausführen, fallen Kosten an. Weitere Informationen finden Sie unter Amazon CloudWatch – Preise.

   Contributor Insights analysiert nur neue Protokollereignisse, nachdem eine Regel erstellt wurde. Eine Regel kann keine Log-Ereignisse verarbeiten, die zuvor von CloudWatch Logs verarbeitet wurden.
10. (Optional) Fügen Sie unter Tags ein oder mehrere Schlüssel/Wert-Paare als Tags für diese Gruppe hinzu. Mithilfe von Stichwörtern können Sie Ihre AWS Ressourcen identifizieren und organisieren und Ihre AWS Kosten verfolgen. Weitere Informationen finden Sie unter Verschlagworten Sie Ihre Amazon-Ressourcen CloudWatch .
11. Wählen Sie Erstellen.

So erstellen Sie eine Regel von Grund auf neu:

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Contributor Insights aus.
3. Wählen Sie Regel erstellen aus.
4. Wählen Sie für Log group(s) (Protokollgruppe(n)) die Protokollgruppe(n) aus, die von der Regel überwacht werden sollen. Sie können bis zu 20 Protokollgruppen auswählen. Wenn Sie mit einem Überwachungskonto angemeldet sind, das für CloudWatch kontenübergreifende Beobachtung eingerichtet ist, können Sie Protokollgruppen in Quellkonten auswählen und auch eine einzige Regel einrichten, um Protokollgruppen in verschiedenen Konten zu analysieren.

   - (Optional) Um alle Protokollgruppen auszuwählen, die Namen haben, die mit einer bestimmten Zeichenfolge beginnen, wählen Sie aus dem Dropdown-Menü Select by prefix match (Nach Präfixübereinstimmung auswählen) aus und geben Sie dann das Präfix ein.

> **Note**
>
> Für jedes Protokollereignis, das Ihrer Regel entspricht, fallen Gebühren an. Wenn Sie aus dem Dropdown-Menü Select by prefix match (Nach Präfixübereinstimmung auswählen) auswählen, sollten Sie sich bewusst sein, mit wie vielen Protokollgruppen das Präfix übereinstimmen kann. Wenn Sie mehr Protokollgruppen suchen, als Sie beabsichtigten, können unerwartete Gebühren anfallen. Weitere Informationen finden Sie unter CloudWatch Amazon-Preise.

5. Wählen Sie für Rule type (Regltyp) Custom rule (Standardregel) aus.
6. Wählen Sie für Protokollformat die Option JSON oder CLF aus.
7. Sie können die Erstellung der Regel über den Assistenten abschließen oder die Registerkarte Syntax auswählen und die Regelsyntax manuell angeben.

   Gehen Sie wie folgt vor, um den Assistenten weiter zu verwenden:

   a. Geben Sie unter Contribution (Beitrag), Key (Schlüssel) einen Contributor-Typ ein, über den Sie einen Bericht wünschen. Der Bericht zeigt die Top-N-Werte für diesen Contributor-Typ an.

      Gültige Einträge sind alle Protokollfelder mit Werten. Beispiele hierfür sind **`requestId`**, **`sourceIPaddress`** und **`containerID`**.

      Informationen zur Suche nach den Namen der Protokollfelder für die Protokolle in einer bestimmten Protokollgruppe finden Sie unter Protokollfelder suchen.

      Schlüssel, die größer als 1 KB sind, werden auf 1 KB gekürzt.

   b. (Optional) Wählen Sie Add new key (Neuen Schlüssel hinzufügen), um weitere Schlüssel hinzuzufügen. Sie können bis zu vier Schlüssel in eine Regel einfügen. Wenn Sie mehrere Schlüssel eingeben, werden die Contributors im Bericht durch eindeutige Wertkombinationen der Schlüssel definiert. Wenn Sie beispielsweise drei Schlüssel angeben, wird jede eindeutige Kombination von Werten für die drei Schlüssel als eindeutiger Contributor gezählt.

   c. (Optional) Wenn Sie einen Filter hinzufügen möchten, um den Umfang Ihrer Ergebnisse einzuschränken, wählen Sie Add filter (Filter hinzufügen) aus. Geben Sie unter Übereinstimmung den Namen des Protokollfelds ein, nach dem Sie filtern möchten. Wählen

Sie dann für Bedingung einen Vergleichsoperator aus und geben Sie einen Wert ein, nach dem Sie filtern möchten.

Sie können einer Regel bis zu vier Filter hinzufügen. Mehrere Filter werden durch die UND-Logik verbunden, so dass nur Protokollereignisse ausgewertet werden, die mit allen Filtern übereinstimmen.

> **Note**
>
> Arrays, die Vergleichsoperatoren folgen, etwa `In`, `NotIn` oder `StartsWith`, können bis zu 10 Zeichenfolgenwerte enthalten. Weitere Informationen zur Syntax von Contributor Insights-Regeln finden Sie unter Contributor Insights-Regelsyntax.

d. Wählen Sie für Aggregieren nach die Option Count oder Sum aus. Die Auswahl von Count bewirkt, dass die Contributor-Rangfolge auf der Anzahl der Vorkommen basiert. Wenn Sie SUM wählen, basiert die Rangfolge auf der aggregierten Summe der Werte des Feldes, das Sie für Contribution (Beitrag), Wert angeben.

8. Gehen Sie folgendermaßen vor, um Ihre Regel als JSON-Objekt einzugeben, anstatt den Assistenten zu verwenden:

   a. Wählen Sie die Registerkarte Syntax aus.

   b. Geben Sie im Regeltext das JSON-Objekt für Ihre Regel ein. Hinweise zur Regelsyntax finden Sie unter Contributor Insights-Regelsyntax.

9. Wählen Sie Weiter aus.

10. Geben Sie unter Rule name (Name der Regel) einen Namen ein. Gültige Zeichen sind A-Z, a-z, 0-9, "-", "_" und ".".

11. Wählen Sie aus, ob die Regel in einem deaktivierten oder aktivierten Zustand erstellt werden soll. Wenn Sie sie aktivieren, beginnt sie sofort mit der Analyse Ihrer Daten. Wenn Sie aktivierte Regeln ausführen, fallen Kosten an. Weitere Informationen finden Sie unter Amazon CloudWatch – Preise.

    Contributor Insights analysiert nur neue Protokollereignisse, nachdem eine Regel erstellt wurde. Eine Regel kann keine Log-Ereignisse verarbeiten, die zuvor von CloudWatch Logs verarbeitet wurden.

12. (Optional) Fügen Sie unter Tags ein oder mehrere Schlüssel/Wert-Paare als Tags für diese Gruppe hinzu. Mithilfe von Stichwörtern können Sie Ihre AWS Ressourcen identifizieren

und organisieren und Ihre AWS Kosten verfolgen. Weitere Informationen finden Sie unter Verschlagworten Sie Ihre Amazon-Ressourcen CloudWatch .

13. Wählen Sie Weiter.
14. Bestätigen Sie die eingegebenen Einstellungen und wählen Sie Create rule (Regel erstellen).

Sie können Regeln, die Sie erstellt haben, deaktivieren, aktivieren oder löschen.

So aktivieren, deaktivieren oder löschen Sie eine Regel in Contributor Insights

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Contributor Insights aus.
3. Markieren Sie in der Liste der Regeln auf der linken Seite das Kontrollkästchen neben einer einzelnen Regel.

   Integrierte Regeln werden von AWS Diensten erstellt und können nicht bearbeitet, deaktiviert oder gelöscht werden.
4. Wählen Sie Actions (Aktionen) und dann die gewünschte Option aus.

Suchen von Protokollfeldern

Wenn Sie eine Regel erstellen, müssen Sie die Namen der Felder in den Protokolleinträgen in einer Protokollgruppe kennen.

So suchen Sie die Protokollfelder in einer Protokollgruppe

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich unter Logs, (Protokolle) die Option Insights(Einblicke) aus.
3. Wählen Sie oberhalb des Abfrage-Editors eine oder mehrere Protokollgruppen aus, die abgefragt werden sollen.

   Wenn Sie eine Protokollgruppe auswählen, erkennt CloudWatch Logs Insights automatisch Felder in den Daten in der Protokollgruppe und zeigt sie im rechten Bereich unter Entdeckte Felder an.

Contributor Insights-Regelsyntax

In diesem Abschnitt wird die Syntax für Contributor Insights-Regeln erläutert. Verwenden Sie diese Syntax nur, wenn Sie eine Regel erstellen, indem Sie einen JSON-Block eingeben. Wenn Sie den Assistenten zum Erstellen einer Regel verwenden, müssen Sie die Syntax nicht kennen. Weitere Hinweise zum Erstellen von Regeln mit dem Assistenten finden Sie unter Eine Contributor-Insights-Regel erstellen.

Bei der Übereinstimmung von Regeln zum Protokollieren von Ereignisfeldnamen und -werten wird zwischen Groß- und Kleinschreibung unterschieden.

Das folgende Beispiel veranschaulicht die Syntax für JSON-Protokolle.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "API-Gateway-Access-Logs*",
        "Log-group-name2"
    ],
    "LogFormat": "JSON",
    "Contribution": {
        "Keys": [
            "$.ip"
        ],
        "ValueOf": "$.requestBytes",
        "Filters": [
            {
                "Match": "$.httpMethod",
                "In": [
                    "PUT"
                ]
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Felder in Contributor Insights-Regeln

Schema

Der Wert von `Schema` für eine Regel, die CloudWatch Protokolldaten analysiert, muss immer `{"Name": "CloudWatchLogRule", "Version": 1}`

LogGroupNames

Ein Array von Zeichenfolgen. Für jedes Element im Array können Sie optional `*` am Ende einer Zeichenfolge verwenden, um alle Protokollgruppen mit Namen einzuschließen, die mit diesem Präfix beginnen.

Seien Sie vorsichtig bei der Verwendung von Platzhaltern mit Protokollgruppennamen. Für jedes Protokollereignis, das einer Regel entspricht, fallen Gebühren an. Wenn Sie versehentlich mehr Protokollgruppen suchen, als Sie beabsichtigten, können unerwartete Gebühren anfallen. Weitere Informationen finden Sie unter CloudWatch Amazon-Preise.

LogGroupARNs

Wenn Sie diese Regel in einem CloudWatch kontoübergreifenden Observability-Monitoring-Konto erstellen, können Sie `LogGroupARNs` damit Log-Gruppen in Quellkonten angeben, die mit dem Monitoring-Konto verknüpft sind, und Log-Gruppen im Monitoring-Konto selbst angeben. Sie müssen entweder `LogGroupNames` oder `LogGroupARNs` in Ihrer Regel angeben, aber nicht beides.

`LogGroupARNs` ist ein Array von Zeichenfolgen. Für jedes Element im Array können Sie es in bestimmten Situationen optional `*` als Platzhalter verwenden. Sie können zum Beispiel `arn:aws:logs:us-west-1:*:log-group/MyLogGroupName2` angeben, um Protokollgruppen mit dem Namen `MyLogGroupName2` in allen Quellkonten und im Überwachungskonto in der Region USA West (Nordkalifornien) festzulegen. Sie können auch `arn:aws:logs:us-west-1:111122223333:log-group/GroupNamePrefix*` angeben, um alle Protokollgruppen in USA-West (Nordkalifornien) in 111122223333 anzugeben, deren Namen mit `GroupNamePrefix` beginnen.

Sie können keine unvollständige AWS Konto-ID als Präfix mit einem Platzhalter angeben.

Seien Sie vorsichtig bei der Verwendung von Platzhaltern mit Protokollgruppen-ARNs. Für jedes Protokollereignis, das einer Regel entspricht, fallen Gebühren an. Wenn Sie versehentlich mehr Protokollgruppen suchen, als Sie beabsichtigten, können unerwartete Gebühren anfallen. Weitere Informationen finden Sie unter CloudWatch Amazon-Preise.

LogFormat

Gültige Werte sind `JSON` und `CLF`.

Contribution (Beitrag)

Dieses Objekt enthält ein `Keys`-Array mit bis zu vier Elementen, optional ein einzelnes `ValueOf` und optional ein Array von bis zu vier `Filters`.

Schlüssel

Ein Array mit bis zu vier Protokollfeldern, die als Dimensionen zum Klassifizieren von Contributors verwendet werden. Wenn Sie mehrere Schlüssel eingeben, wird jede eindeutige Kombination von Werten für die Schlüssel als eindeutiger Contributor gezählt. Die Felder müssen mit der JSON-Eigenschaftsformatnotation angegeben werden.

ValueOf

(Optional) Geben Sie dies nur an, wenn Sie `Sum` als Wert von `AggregateOn` angeben. `ValueOf` gibt ein Protokollfeld mit numerischen Werten an. Bei diesem Regeltyp werden die Contributors nach ihrer Summe des Wertes dieses Feldes und nicht nach ihrer Anzahl von Vorkommen in den Protokolleinträgen geordnet. Wenn Sie beispielsweise Contributors nach ihrem `BytesSent`-Gesamtwert über einen Zeitraum hinweg sortieren möchten, würden Sie `ValueOf` auf `BytesSent` setzen und `Sum` für `AggregateOn` angeben.

Filter

(Optional) Gibt ein Array von bis zu vier Filtern an, um die Protokollereignisse einzuschränken, die im Bericht enthalten sind. Wenn Sie mehrere Filter angeben, wertet Contributor Insights diese mit einem logischen AND-Operator aus. Sie können hiermit irrelevante Protokollereignisse aus Ihrer Suche herausfiltern. Alternativ können Sie einen einzelnen Contributor auswählen, um dessen Verhalten zu analysieren.

Jedes Element im Array muss ein `Match`-Feld und ein Feld enthalten, das den Typ des zu verwendenden übereinstimmenden Operators angibt.

Das `Match`-Feld gibt ein Protokollfeld an, das im Filter ausgewertet werden soll. Das Protokollfeld wird mithilfe der JSON-Eigenschaftsformatnotation angegeben.

Das übereinstimmende Operatorfeld muss eines der folgenden sein: `In`, `NotIn`, `StartsWith`, `GreaterThan`, `LessThan`, `EqualTo`, `NotEqualTo` oder `IsPresent`. Wenn das Operatorfeld `In`, `NotIn` oder `StartsWith` ist, folgt ein Array von Zeichenfolgenwerten, auf die überprüft

werden soll. Contributor Insights wertet das Array von Zeichenfolgenwerten mit einem OR-Operator aus. Das Array kann bis zu 10 Zeichenfolgenwerte enthalten.

Wenn das Operatorfeld `GreaterThan`, `LessThan`, `EqualTo` oder `NotEqualTo` ist, folgt ein einzelner numerischer Wert, mit dem verglichen werden soll.

Wenn das Operatorfeld `IsPresent` ist, folgt entweder `true` oder `false`. Dieser Operator gleicht Protokollereignisse basierend daraufhin ab, ob das angegebene Protokollfeld in dem Protokollereignis vorhanden ist. `isPresent` funktioniert nur mit Werten im Blattknoten der JSON-Eigenschaften. Beispielsweise wertet ein Filter, der nach Übereinstimmungen mit `c-count` sucht, kein Protokollereignis mit einem Wert von `details.c-count.c1` aus.

Im Folgenden finden Sie vier Filterbeispiele:

```
{"Match": "$.httpMethod", "In": [ "PUT", ] }
{"Match": "$.StatusCode", "EqualTo": 200 }
{"Match": "$.BytesReceived", "GreaterThan": 10000}
{"Match": "$.eventSource", "StartsWith": [ "ec2", "ecs" ] }
```

AggregateOn

Gültige Werte sind `Count` und `Sum`. Gibt an, ob der Bericht basierend auf der Häufigkeit des Auftretens oder einer Summe der Werte des Feldes aggregiert werden soll, das im `ValueOf`-Feld angegeben ist.

JSON-Eigenschaftsformatnotation

Die Felder `Keys`, `ValueOf` und `Match` folgen dem JSON-Eigenschaftsformat mit Punktnotierung, wobei $ den Stamm des JSON-Objekts darstellt. Danach folgt ein Punkt und dann eine alphanumerische Zeichenfolge mit dem Namen der Untereigenschaft. Mehrere Eigenschaftenebenen werden unterstützt.

Das erste Zeichen der Zeichenfolge kann nur A-Z oder a-z sein. Die folgenden Zeichen der Zeichenfolge können A-Z, a-z oder 0-9 sein.

Die folgende Liste zeigt gültige Beispiele für das JSON-Eigenschaftsformat:

```
$.userAgent
$.endpoints[0]
$.users[1].name
```

```
$.requestParameters.instanceId
```

Zusätzliches Feld in Regeln für CLF-Protokolle

CLF-Protokollereignisse (Common Log Format) haben keine Namen für die Felder wie JSON. Um die Felder bereitzustellen, die für Contributor Insights-Regeln verwendet werden sollen, kann ein CLF-Protokollereignis als Array mit einem Index beginnend mit 1 behandelt werden. Sie können das erste Feld als **"1"**, das zweite Feld als **"2"** usw. angeben.

Um eine Regel für ein CLF-Protokoll besser lesbar zu machen, können Sie `Fields` verwenden. Auf diese Weise können Sie einen Namensalias für CLF-Feldspeicherorte bereitstellen. Sie können beispielsweise angeben, dass der Speicherort „4" eine IP-Adresse ist. Einmal angegeben, kann `IpAddress` als Eigenschaft in `Keys`, `ValueOf` und `Filters` in der Regel verwendet werden.

Nachfolgend finden Sie ein Beispiel für eine Regel für ein CLF-Protokoll, das das `Fields`-Feld verwendet.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "API-Gateway-Access-Logs*"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "4": "IpAddress",
        "7": "StatusCode"
    },
    "Contribution": {
        "Keys": [
            "IpAddress"
        ],
        "Filters": [
            {
                "Match": "StatusCode",
                "EqualTo": 200
            }
        ]
    },
    "AggregateOn": "Count"
```

```
}
```

Beispiele für Contributor Insights-Regeln

Dieser Abschnitt enthält Beispiele, die Anwendungsfälle für Contributor Insights-Regeln veranschaulichen.

VPC-Flow-Protokolle: Byte-Übertragungen nach Quell- und Ziel-IP-Adresse

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "4": "srcaddr",
        "5": "dstaddr",
        "10": "bytes"
    },
    "Contribution": {
        "Keys": [
            "srcaddr",
            "dstaddr"
        ],
        "ValueOf": "bytes",
        "Filters": []
    },
    "AggregateOn": "Sum"
}
```

VPC-Flow-Protokolle: Höchste Anzahl von HTTPS-Anforderungen

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
```

```
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "5": "destination address",
        "7": "destination port",
        "9": "packet count"
    },
    "Contribution": {
        "Keys": [
            "destination address"
        ],
        "ValueOf": "packet count",
        "Filters": [
            {
                "Match": "destination port",
                "EqualTo": 443
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

VPC-Flow-Protokolle: Abgelehnte TCP-Verbindungen

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "3": "interfaceID",
        "4": "sourceAddress",
        "8": "protocol",
        "13": "action"
    },
    "Contribution": {
        "Keys": [
            "interfaceID",
```

```
            "sourceAddress"
        ],
        "Filters": [
            {
                "Match": "protocol",
                "EqualTo": 6
            },
            {
                "Match": "action",
                "In": [
                    "REJECT"
                ]
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Antworten von Route 53 NxDomain nach Quelladresse

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Count",
    "Contribution": {
        "Filters": [
            {
                "Match": "$.rcode",
                "StartsWith": [
                    "NXDOMAIN"
                ]
            }
        ],
        "Keys": [
            "$.srcaddr"
        ]
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "<loggroupname>"
    ]
```

```
}
```

Auflösungsabfragen von Route 53 nach Domainnnamen

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Count",
    "Contribution": {
        "Filters": [],
        "Keys": [
            "$.query_name"
        ]
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "<loggroupname>"
    ]
}
```

Auflösungsabfragen von Route 53 nach Abfragetyp und Quelladresse

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Count",
    "Contribution": {
        "Filters": [],
        "Keys": [
            "$.query_type",
            "$.srcaddr"
        ]
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "<loggroupname>"
    ]
}
```

Anzeigen von Contributor Insights-Berichten

Gehen Sie folgendermaßen vor, um Diagramme mit Berichtsdaten und eine Rangliste der Contributors anzuzeigen, die von Ihren Regeln gefunden wurden.

So zeigen Sie Regelberichte an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Contributor Insights aus.
3. Wählen Sie in der Liste der Regeln den Namen einer Regel aus.

   Das Diagramm zeigt die Ergebnisse der Regel der letzten drei Stunden an. Die Tabelle unter dem Diagramm zeigt die Top 10-Contributors.
4. Um die Anzahl der in der Tabelle angezeigten Contributors zu ändern, wählen Sie Top 10-Contributors oben im Diagramm aus.
5. Um das Diagramm so zu filtern, dass nur die Ergebnisse eines einzelnen Contributors angezeigt werden, wählen Sie diesen Contributor in der Tabellenlegende aus. Um alle Contributors wieder anzuzeigen, wählen Sie denselben Contributor in der Legende erneut aus.
6. Um den im Bericht angezeigten Zeitraum zu ändern, wählen Sie am oberen Rand des Diagramms 15 Min., 30 Min., 1 Std., 2 Std., 3 Std. oder Benutzerdefiniert aus.

   Der maximale Zeitbereich für den Bericht beträgt 24 Stunden, Sie können jedoch ein 24-Stunden-Fenster auswählen, das vor 15 Tagen aufgetreten ist. Um ein Zeitfenster in der Vergangenheit auszuwählen, klicken Sie auf Benutzerdefiniert, Absolut, und geben Sie dann Ihr Zeitfenster an.
7. Um die Länge des Zeitraums zu ändern, der für die Aggregation und Rangfolge der Contributors verwendet wird, wählen Sie Zeitraum oben im Diagramm aus. Das Anzeigen eines längeren Zeitraums zeigt in der Regel einen gleichmäßigeren Bericht mit wenigen Spitzen. Bei Auswahl eines kürzeren Zeitraums ist die Wahrscheinlichkeit größer, dass Spitzen angezeigt werden.
8. Um dieses Diagramm zu einem CloudWatch Dashboard hinzuzufügen, wählen Sie Zum Dashboard hinzufügen.
9. Um das CloudWatch Logs Insights-Abfragefenster zu öffnen, in dem die Protokollgruppen in diesem Bericht bereits im Abfragefeld geladen sind, wählen Sie Protokolle anzeigen.
10. Um die Berichtsdaten in die Zwischenablage oder eine CSV-Datei zu exportieren, wählen Sie Exportieren aus.

Grafisches Darstellen von durch Regeln generierte Metriken

Contributor Insights bietet eine Metrikberechnungsfunktion, `INSIGHT_RULE_METRIC`. Sie können diese Funktion verwenden, um Daten aus einem Contributor Insights-Bericht zu einem Diagramm auf der Registerkarte Metriken der CloudWatch Konsole hinzuzufügen. Sie können auch einen Alarm basierend auf dieser mathematischen Funktion einstellen. Weitere Informationen zu Metrikberechnungsfunktionen finden Sie unter Verwenden von Metrikberechnungen

Um diese Metrikberechnungsfunktion verwenden zu können, müssen Sie an einem Konto angemeldet sein, das über die Berechtigungen `cloudwatch:GetMetricData` und `cloudwatch:GetInsightRuleReport` verfügt.

Die Syntax lautet `INSIGHT_RULE_METRIC(`*`ruleName`*`, `*`metricName`*`)`. *`ruleName`* ist der Name einer Contributor Insights-Regel und *`metricName`* ist einer der Werte in der folgenden Liste. Der Wert von *`metricName`* bestimmt, welche Art von Daten die mathematische Funktion zurückgibt.

- `UniqueContributors` – die Anzahl der eindeutigen Mitwirkenden für jeden Datenpunkt.
- `MaxContributorValue` – der Wert des obersten Mitwirkenden für jeden Datenpunkt. Die Identität des Contributors kann sich für jeden Datenpunkt in dem Diagramm ändern.

  Wenn diese Regel nach `Count` aggregiert wird, ist der Top-Contributor für jeden Datenpunkt der Contributor mit den meisten Vorkommen in diesem Zeitraum. Wenn die Regel nach `Sum` aggregiert wird, ist der Top-Contributor der Contributor mit der größten Summe in dem durch das `Value` der Regel angegebenen Protokollfeld während dieses Zeitraums.
- `SampleCount` – Die Anzahl der Datenpunkte, mit denen die Regel übereinstimmt.
- `Sum` – die Summe der Werte aller Mitwirkenden während des durch diesen Datenpunkt repräsentierten Zeitraums.
- `Minimum` – der Mindestwert einer einzelnen Beobachtung während des durch diesen Datenpunkt repräsentierten Zeitraums.
- `Maximum` – der Höchstwert einer einzelnen Beobachtung während des durch diesen Datenpunkt repräsentierten Zeitraums.
- `Average` – der durchschnittliche Wert aller Mitwirkenden während des durch diesen Datenpunkt repräsentierten Zeitraums.

Festlegen eines Alarms für Contributor Insights-Metrikdaten

Mit der Funktion `INSIGHT_RULE_METRIC` können Sie Alarme für Metriken festlegen, die von Contributor Insights generiert werden. Beispielsweise können Sie einen Alarm basierend auf dem Prozentsatz der zurückgewiesenen TCP-Verbindungen (Transmission Control Protocol) erstellen. Um mit dieser Art von Alarm zu beginnen, können Sie Regeln wie die in den folgenden beiden Beispielen gezeigten erstellen:

Beispielregel: "" RejectedConnectionsRule

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "3": "interfaceID",
        "4": "sourceAddress",
        "8": "protocol",
        "13": "action"
    },
    "Contribution": {
        "Keys": [
            "interfaceID",
            "sourceAddress"
        ],
        "Filters": [
            {
                "Match": "protocol",
                "EqualTo": 6
            },
            {
                "Match": "action",
                "In": [
                    "REJECT"
                ]
            }
        ]
    },
```

```
    "AggregateOn": "Sum"
}
```

Beispielregel: "TotalConnectionsRule"

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "3": "interfaceID",
        "4": "sourceAddress",
        "8": "protocol",
        "13": "action"
    },
    "Contribution": {
        "Keys": [
            "interfaceID",
            "sourceAddress"
        ],
        "Filters": [
            {
                "Match": "protocol",
                "EqualTo": 6
            }
    "AggregateOn": "Sum"
}
```

Nachdem Sie Ihre Regeln erstellt haben, können Sie in der CloudWatch Konsole die Registerkarte „Metriken“ auswählen. Dort können Sie die folgenden mathematischen Beispielausdrücke verwenden, um die von Contributor Insights gemeldeten Daten grafisch darzustellen:

Example: Metric math expressions (Beispiel: Metrikberechnungs-Ausdrücke)

```
e1 INSIGHT_RULE_METRIC("RejectedConnectionsRule", "Sum")
e2 INSIGHT_RULE_METRIC("TotalConnectionsRule", "Sum")
e3 (e1/e2)*100
```

Im Beispiel werden mit dem Metrikberechnungs-Ausdruck `e3` alle abgelehnten TCP-Verbindungen zurückgegeben. Wenn Sie benachrichtigt werden möchten, wenn 20 % der TCP-Verbindungen abgelehnt werden, können Sie den Schwellenwert im Ausdruck von `100` zu `20` ändern.

> **Note**
>
> Für eine Metrik, die Sie überwachen, können Sie im Abschnitt Metrics (Metriken) einen Alarm erstellen. Auf der Registerkarte Graphed metrics (Grafisch dargestellt Metriken) können Sie das Create alarm (Alarm erstellen)-Symbol unter der Spalte Actions (Aktionen) auswählen. Das Symbol für Alarm erstellen sieht aus wie eine Glocke.

Weitere Hinweise zur grafischen Darstellung von Metriken und zum Verwenden von Metrikberechnungsfunktionen finden Sie im folgenden Abschnitt: Fügen Sie einem CloudWatch Diagramm einen mathematischen Ausdruck hinzu.

Verwenden von integrierten Regeln für Contributor Insights

Sie können die integrierten Regeln von Contributor Insights verwenden, um Metriken aus anderen AWS Diensten zu analysieren. Die folgenden Dienste unterstützen integrierte Regeln:

- Contributor Insights für Amazon DynamoDB im Entwicklerhandbuch für Amazon DynamoDB.
- Verwenden von integrierten Regeln für Contributor Insights im AWS PrivateLink -Handbuch.

Einblicke in CloudWatch Amazon-Anwendungen

Amazon CloudWatch Application Insights erleichtert die Beobachtbarkeit Ihrer Anwendungen und der zugrunde liegenden AWS Ressourcen. Es kann Ihnen helfen, die besten Überwachungen für Ihre Anwendungsressourcen einzurichten, um Daten kontinuierlich auf Anzeichen von Problemen mit Ihren Anwendungen zu analysieren. Application Insights, das auf SageMakerund anderen AWS Technologien basiert, bietet automatisierte Dashboards, die potenzielle Probleme mit überwachten Anwendungen aufzeigen, sodass Sie aktuelle Probleme mit Ihren Anwendungen und Ihrer Infrastruktur schnell isolieren können. Die verbesserte Transparenz zum Status Ihrer Anwendungen mit Application Insights kann Ihnen helfen, Ihre durchschnittliche Reparaturzeit (mean time to repair, MTTR) zu reduzieren, um Ihre Anwendungsprobleme zu beheben.

Wenn Sie Ihre Anwendungen zu Amazon CloudWatch Application Insights hinzufügen, scannt es die Ressourcen in den Anwendungen und empfiehlt und konfiguriert Metriken und meldet sich

CloudWatchfür Anwendungskomponenten an. BeispielAnwendungskomponenten können SQL-Server-Backend-Datenbanken und Microsoft IIS/Web-Tiers sein. Application Insights analysiert metrische Muster anhand historischer Daten, um Anomalien zu erkennen, und erkennt kontinuierlich Fehler und Ausnahmen aus Ihren Anwendungen, -Fehlerprotokoll und IIS. Es korreliert diese Beobachtungen mit einer Kombination aus Klassifikationsalgorithmen und integrierten Regeln. Dann erstellt es automatisch Dashboards, die die relevanten Beobachtungen und Informationen zur Problemschwere anzeigen, um Ihnen zu helfen, Ihre Aktionen zu priorisieren. Für häufige Probleme in .NET- und SQL-Anwendungs-Stacks, wie z. B. Anwendungslatenz, fehlgeschlagene SQL Server-Sicherungen, Speicherlecks, große HTTP-Anfragen und abgebrochene I/O-Operationen, bietet es zusätzliche Einblicke, die auf eine mögliche Ursache und Schritte zur Behebung hinweisen. Die integrierte Integration mit AWS SSM OpsCenter ermöglicht es Ihnen, Probleme zu lösen, indem Sie das entsprechende Systems Manager Automation-Dokument ausführen.

Sections

Was ist Amazon CloudWatch Application Insights?

CloudWatch Application Insights unterstützt Sie bei der Überwachung Ihrer Anwendungen, die Amazon EC2 EC2-Instances zusammen mit anderen Anwendungsressourcen verwenden. Es identifiziert Schlüsselmetriken, Protokolle und Alarme und richtet diese für Ihre Anwendungsressourcen und Ihren Technologie-Stack ein (z. B. Ihre Microsoft SQL Server-Datenbank, Web (IIS)- und Anwendungsserver, Betriebssystem, Load Balancer und

Warteschlangen). Es überwacht kontinuierlich Metriken und Protokolle, um Anomalien und Fehler zu erkennen und zu korrelieren. Wenn Fehler und Anomalien erkannt werden, generiert Application Insights CloudWatch Ereignisse, anhand derer Sie Benachrichtigungen einrichten oder Maßnahmen ergreifen können. Um die Fehlersuche zu erleichtern, erstellt es automatisierte Dashboards für die erkannten Probleme, die korrelierten Metrikanomalien und Protokollfehler sowie zusätzliche Erkenntnisse, die Sie auf die mögliche Ursache hinweisen. Die automatisierten Dashboards helfen Ihnen dabei, schnell Abhilfemaßnahmen zu treffen, um Ihre Anwendungen funktionstüchtig zu halten und Auswirkungen auf die Endbenutzer Ihrer Anwendung zu vermeiden. Es erstellt auch, OpsItems sodass Sie Probleme mithilfe AWS von SSM lösen können. OpsCenter

Sie können wichtige Leistungsindikatoren wie gespiegelte Schreibtransaktionen/Sekunde, Länge der Wiederherstellungswarteschlange und Transaktionsverzögerung sowie Windows-Ereignisprotokolle konfigurieren. CloudWatch Wenn ein Failover-Ereignis oder ein Problem mit Ihrer SQL HA-Arbeitslast auftritt, z. B. ein eingeschränkter Zugriff auf die Abfrage einer Zieldatenbank, bietet Application Insights automatisierte Einblicke. CloudWatch

CloudWatch Application Insights lässt sich integrieren AWS Launch Wizard, um die Einrichtung der Überwachung mit nur einem Klick für die Bereitstellung von SQL Server-HA-Workloads zu ermöglichen. AWS Wenn Sie in der Launch Wizard Wizard-Konsole die Option zur Einrichtung von Monitoring und Insights mit Application Insights auswählen, richtet CloudWatch Application Insights automatisch relevante Metriken, Protokolle und Alarme ein und beginnt mit der Überwachung neu bereitgestellter Workloads. CloudWatch Auf der Konsole können Sie automatische Einblicke und erkannte Probleme sowie den Zustand Ihrer SQL Server HA-Workloads einsehen. CloudWatch

Inhalt

Funktionen

Application Insights bietet die folgenden Features:

Automatisches Einrichten der Überwachung für Anwendungsressourcen

CloudWatch Application Insights reduziert den Zeitaufwand für die Einrichtung der Überwachung Ihrer Anwendungen. Zu diesem Zweck werden Ihre Anwendungsressourcen gescannt, eine anpassbare Liste mit empfohlenen Metriken und Protokollen bereitgestellt und diese eingerichtet, CloudWatch um den erforderlichen Einblick in Ihre Anwendungsressourcen wie Amazon EC2 und Elastic Load Balancers (ELB) zu bieten. Der Service richtet außerdem dynamische Alarme für überwachte Metriken ein. Die Alarme werden automatisch aktualisiert, basierend auf den in den letzten zwei Wochen festgestellten Anomalien.

Problemerkennung und Benachrichtigung

CloudWatch Application Insights erkennt Anzeichen potenzieller Probleme mit Ihrer Anwendung, wie z. B. metrische Anomalien und Protokollfehler. Es korreliert diese Beobachtungen mit möglichen Problemen mit Ihrer Anwendung. Anschließend generiert es CloudWatch Ereignisse, die so konfiguriert werden können, dass sie Benachrichtigungen empfangen oder Maßnahmen ergreifen. Dadurch entfällt die Notwendigkeit, individuelle Alarme bei Metriken oder Protokollfehlern zu erstellen.

Fehlerbehebung

CloudWatch Application Insights erstellt CloudWatch automatische Dashboards für erkannte Probleme. Die Dashboards zeigen Details zum Problem, einschließlich der zugehörigen Metrikanomalien und Protokollfehler, die Ihnen bei der Fehlersuche helfen. Sie liefern außerdem zusätzliche Erkenntnisse, die auf mögliche Ursachen für die Anomalien und Fehler hinweisen.

Konzepte

Die folgenden Konzepte sind wichtig, um zu verstehen, wie Application Insights Ihre Anwendung überwacht.

Komponente

Eine automatisch gruppierte, eigenständige oder benutzerdefinierte Gruppierung ähnlicher Ressourcen, aus denen eine Anwendung besteht. Wir empfehlen, ähnliche Ressourcen zur besseren Überwachung in benutzerdefinierten Komponenten zusammenzufassen.

Beobachtung

Ein einzelnes Ereignis (Metrikanomalie, Protokollfehler oder Ausnahme), das mit einer Anwendung oder Anwendungsressource erkannt wird.

Problem

Probleme werden durch Korrelation, Klassifizierung und Gruppierung von Beobachtungen erkannt.

Definitionen anderer Schlüsselkonzepte für CloudWatch Application Insights finden Sie unter Amazon CloudWatch Concepts.

Preisgestaltung

CloudWatch Application Insights richtet empfohlene Metriken und Protokolle für ausgewählte Anwendungsressourcen ein und verwendet CloudWatch Metriken, Protokolle und Ereignisse für Benachrichtigungen zu erkannten Problemen. Diese Funktionen werden Ihrem AWS Konto entsprechend der CloudWatch Preisgestaltung in Rechnung gestellt. Bei erkannten Problemen OpsItems werden auch SSM von Application Insights erstellt, um Sie über Probleme zu informieren. Darüber hinaus erstellt Application Insights SSM-Parameter Store-Parameter, um die CloudWatch Agenten auf Ihren Instances zu konfigurieren. Die Features von Amazon EC2 Systems Manager werden gemäß der SSM-Preisgestaltung berechnet. Es entstehen Ihnen keine Kosten für Einrichtungshilfe, die Überwachung der Datenanalyse oder die Problemerkennung.

Kosten für CloudWatch Application Insights

Die Kosten für Amazon EC2 beinhalten die Nutzung der folgenden Features:

- CloudWatch Agent
  - CloudWatch Agent-Protokollgruppen
  - CloudWatch Agent-Metriken
  - Prometheus-Protokollgruppen (für JMX-Workloads)

Die Kosten für alle Ressourcen beinhalten die Nutzung der folgenden Features:

- CloudWatch Alarme (Großteil der Kosten)
- SSM OpsItems (minimale Kosten)

Beispiel für Kostenberechnung

Die Kosten in diesem Beispiel werden gemäß dem folgenden Szenario betrachtet.

Sie haben eine Ressourcengruppe erstellt, die Folgendes beinhaltet:

- Eine Amazon-EC2-Instance mit installiertem SQL-Server.

- Ein angehängtes Amazon-EBS-Volume.

Wenn Sie diese Ressourcengruppe mit CloudWatch Application Insights verbinden, wird der auf der Amazon EC2 EC2-Instance installierte SQL Server-Workload erkannt. CloudWatch Application Insights beginnt mit der Überwachung der folgenden Metriken.

Die folgenden Metriken werden für die SQL-Server-Instance überwacht:

- CPUUtilization
- StatusCheckFailed
- Speicher % übertragene Bytes im Gebrauch
- Verfügbarer Speicher Mbytes
- Netzwerkschnittstellen-Bytes gesamt/Sekunde
- Auslagerungsdatei % Verwendung
- Physische Festplatte % Festplattenzeit
- Processor % Processor Time
- SQLServer: Puffer-Manager-Cache-Trefferquote
- SQLServer:Lebenserwartung des Puffer-Managers
- SQLServer:General Statistics Processes blocked
- SQLServer:General Statistics User Connections
- SQLServer:Locks Number of Deadlocks/sec
- SQLServer:SQL Statistics Batch Requests/sec
- System Processor Queue Length

Die folgenden Metriken werden für die Volumes überwacht, die mit der SQL-Server-Instance verbunden sind:

- VolumeReadBytes
- VolumeWriteBytes
- VolumeReadOps
- VolumeWriteOps
- VolumeTotalReadTime

- VolumeTotalWriteTime
- VolumeIdleTime
- VolumeQueueLength
- VolumeThroughputPercentage
- VolumeConsumedReadWriteOps
- BurstBalance

Für dieses Szenario werden die Kosten auf der CloudWatch Preisseite und der SSM-Preisseite berechnet:

- Benutzerdefinierte Metriken

  In diesem Szenario werden 13 der oben genannten Messwerte an die CloudWatch Nutzung des CloudWatch Agenten ausgegeben. Diese Metriken werden als benutzerdefinierte Metriken behandelt. Die Kosten für jede benutzerdefinierte Metrik betragen 0,30 USD/Monat. Die Gesamtkosten für diese benutzerdefinierten Metriken betragen 13 x 0,30 USD = 3,90 USD pro Monat.
- Alarme

  Für dieses Szenario überwacht CloudWatch Application Insights insgesamt 26 Metriken, wodurch 26 Alarme generiert werden. Die Kosten für jeden Alarm betragen 0,10 USD/Monat. Die Gesamtkosten für Alarme betragen 26 x 0,10 USD = 2,60 USD pro Monat.
- Datenerfassung und Fehlerprotokolle

  Die Kosten für die Datenerfassung betragen 0,05 USD pro GB und der Speicherplatz für das SQL-Server-Fehlerprotokoll beträgt 0,03 USD pro GB. Die Gesamtkosten für die Datenerfassung und das Fehlerprotokoll betragen 0,05 USD pro GB + 0,03 USD pro GB = 0,08 USD pro GB.
- Amazon EC2 Systems Manager OpsItems

  Für jedes von CloudWatch Application OpsItem Insights erkannte Problem wird ein SSM erstellt. Für Anzahl n an Problemen in Ihrer Anwendung belaufen sich die Gesamtkosten auf 0,00267 USD x n /Monat.

Zugehörige Services

Die folgenden Dienste werden zusammen mit CloudWatch Application Insights verwendet:

Verwandte AWS Dienste

- Amazon CloudWatch bietet systemweiten Einblick in die Ressourcennutzung, die Anwendungsleistung und den Betriebszustand. Es sammelt und verfolgt Metriken, sendet Alarmbenachrichtigungen, aktualisiert die Ressourcen, die Sie überwachen, automatisch auf der Grundlage der von Ihnen definierten Regeln und ermöglicht es Ihnen, Ihre eigenen benutzerdefinierten Messwerte zu überwachen. CloudWatch Application Insights wird CloudWatch insbesondere über die CloudWatch standardmäßigen operativen Dashboards initiiert. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.
- CloudWatch Container Insights sammelt, aggregiert und fasst Metriken und Protokolle aus Ihren containerisierten Anwendungen und Microservices zusammen. Sie können Container Insights verwenden, um Amazon ECS, Amazon Elastic Kubernetes Service und Kubernetes-Plattformen auf Amazon EC2 zu überwachen. Wenn Application Insights auf den Container-Insights- oder Application-Insights-Konsolen aktiviert ist, zeigt Application Insights erkannte Probleme in Ihrem Container-Insights-Dashboard an. Weitere Informationen finden Sie unter Container Insights.
- Amazon DynamoDB ist ein vollständig verwalteter NoSQL-Datenbankservice, mit dem Sie den Verwaltungsaufwand für den Betrieb und die Skalierung verteilter Datenbanken verringern, sodass Sie sich nicht um Hardwarebereitstellung, Einrichtung und Konfiguration, Replikation, Software-Patching oder Cluster-Skalierung kümmern müssen. DynamoDB bietet auch Verschlüsselung im Ruhezustand, wodurch der Betriebsaufwand und die Komplexität, die mit dem Schutz sensibler Daten verbunden sind, eliminiert werden.
- Amazon EC2 bietet skalierbare Rechenkapazität in der AWS Cloud. Mit Amazon EC2 können Sie so viele oder so wenige virtuelle Server starten, wie Sie benötigen, die Sicherheit und das Netzwerk konfigurieren und den Speicher verwalten. Sie können nach hoch oder runter skalieren, um Anforderungsänderungen oder Nutzungsspitzen zu bewältigen, was Ihren Bedarf an Traffic-Prognosen reduziert. Weitere Informationen finden Sie im Amazon EC2-Benutzerhandbuch für Linux-Instances oder Amazon EC2-Handbuch für Windows-Instances.
- Amazon Elastic Block Store (Amazon EBS) bietet Volumes für die Speicherung auf Blockebene, die in Verbindung mit Amazon-EC2-Instances verwendet werden. Amazon-EBS-Volumes verhalten sich wie unformatierte Blockgeräte. Sie können diese Volumes als Geräte auf Ihren Instances mounten. Amazon-EBS-Volumes, die einer Instance angefügt sind, werden als Speicher-Volumes bereitgestellt, die unabhängig von der Instance-Lebenszeit bestehen. Sie können ein Dateisystem über diesen Volumes erstellen oder sie auf dieselbe Art wie ein Blockgerät verwenden (so wie eine Festplatte). Sie können die Konfiguration eines Volumes, das einer Instance zugeordnet ist, dynamisch ändern. Weitere Informationen finden Sie im Benutzerhandbuch für Amazon EBS.

- Amazon EC2 Auto Scaling hilft Ihnen sicherzustellen, dass Sie die richtige Anzahl von EC2-Instances zur Verfügung haben, um die Auslastung Ihrer Anwendung zu bewältigen. Weitere Informationen hierzu finden Sie im Amazon EC2 Auto Scaling-Benutzerhandbuch.
- Elastic Load Balancingverteilt eingehende Anwendungen oder Netzwerkverkehr auf mehrere Ziele, wie EC2-Instances, Container und IP-Adressen, in mehreren Availability Zones. Weitere Informationen finden Sie im Elastic Load Balancing-Benutzerhandbuch.
- IAM ist ein Webservice, mit dem Sie den Zugriff Ihrer Benutzer auf AWS Ressourcen sicher kontrollieren können. Verwenden Sie IAM, um zu kontrollieren, wer Ihre AWS Ressourcen nutzen kann (Authentifizierung), und um zu kontrollieren, welche Ressourcen sie verwenden können und wie sie sie verwenden können (Autorisierung). Weitere Informationen finden Sie unter Authentifizierung und Zugriffskontrolle für Amazon CloudWatch.
- AWS Lambdaermöglicht es Ihnen, serverlose Anwendungen zu erstellen, die aus Funktionen bestehen, die durch Ereignisse ausgelöst werden, und diese mithilfe von CodePipeline und AWS CodeBuild automatisch bereitzustellen. Weitere Informationen finden Sie unter AWS Lambda - Anwendungen.
- AWS Launch Wizard for SQL Server reduziert die Zeit, die für die Bereitstellung der SQL Server-Hochverfügbarkeitslösung in der Cloud benötigt wird. Sie geben Ihre Anwendungsanforderungen, einschließlich Leistung, Anzahl der Knoten und Konnektivität, in die Servicekonsole ein und ermitteln AWS Launch Wizard die richtigen AWS Ressourcen für die Bereitstellung und Ausführung Ihrer SQL Server Always On-Anwendung.
- AWS Resource Groups helfen Ihnen dabei, die Ressourcen zu organisieren, aus denen Ihre Anwendung besteht. Mit Resource Groups können Sie Aufgaben für eine große Anzahl von Ressourcen gleichzeitig verwalten und automatisieren. Nur eine Ressourcengruppe kann für eine einzelne Anwendung registriert werden. Weitere Informationen finden Sie im AWS -Resource-Groups-Benutzerhandbuch.
- Amazon SQS bietet eine sichere, dauerhafte und verfügbare gehostete Warteschlange, die es Ihnen ermöglicht, verteilte Softwaresysteme und -komponenten zu integrieren und zu entkoppeln. Weitere Informationen finden Sie im Amazon SQS-Benutzerhandbuch.
- AWS Step Functionsist ein serverloser Function Composer, mit dem Sie eine Vielzahl von AWS Diensten und Ressourcen, einschließlich AWS Lambda Funktionen, in strukturierten, visuellen Workflows sequenzieren können. Weitere Informationen finden Sie im AWS Step Functions - Benutzerhandbuch.
- AWS SSM OpsCenter aggregiert und standardisiert OpsItems alle Dienste und stellt gleichzeitig kontextbezogene Untersuchungsdaten zu den einzelnen OpsItem, verwandten und verwandten Ressourcen bereit. OpsItems OpsCenter stellt außerdem Systems Manager Automation-

Dokumente (Runbooks) bereit, mit denen Sie Probleme schnell lösen können. Sie können für jedes Objekt durchsuchbare, benutzerdefinierte Daten angeben. OpsItem Sie können auch automatisch generierte Übersichtsberichte OpsItems nach Status und Quelle anzeigen. Weitere Informationen finden Sie im AWS Systems Manager -Benutzerhandbuch.

- Amazon API Gateway ist ein AWS Service für die Erstellung, Veröffentlichung, Wartung, Überwachung und Sicherung von REST, HTTP und WebSocket APIs in jeder Größenordnung. API-Entwickler können APIs erstellen, die auf andere Webdienste sowie auf in der AWS Cloud gespeicherte Daten zugreifen AWS . Weitere Informationen finden Sie im Benutzerhandbuch für Amazon API Gateway.

  > **Note**
  >
  > Application Insights unterstützt nur REST-API-Protokolle (v1 des API-Gateway-Services).

- Amazon Elastic Container Service (Amazon ECS) ist ein vollständig verwalteter Container-Orchestrierungsservice Sie können Amazon ECS verwenden, um Ihre sensibelsten und geschäftskritischsten Anwendungen auszuführen. Weitere Informationen finden Sie im Entwicklerhandbuch zu Amazon Elastic Container Service.
- Amazon Elastic Kubernetes Service (Amazon EKS) ist ein verwalteter Service, mit dem Sie Kubernetes ausführen können, AWS ohne Ihre eigene Kubernetes-Steuerebene oder Knoten installieren, betreiben und warten zu müssen. Kubernetes ist ein Open-Source-System zur Automatisierung der Bereitstellung, Skalierung und Verwaltung von Anwendungen in Containern. Weitere Informationen finden Sie im Amazon-EKS-Benutzerhandbuch.
- Kubernetes auf Amazon EC2. Kubernetes ist eine Open-Source-Software, mit der Sie containerisierte Anwendungen in großem Umfang bereitstellen und verwalten können. Kubernetes verwaltet Cluster von Amazon-EC2-Computing-Instances und führt Container auf diesen Instances mit Prozessen für Bereitstellung, Wartung und Skalierung aus. Mit Kubernetes können Sie jede Art von containerisierten Anwendungen mit demselben Toolset On-Premises und in der Cloud ausführen. Weitere Informationen finden Sie in der Kubernetes-Dokumentation: Erste Schritte.
- Amazon FSx hilft Ihnen, gängige Dateisysteme zu starten und auszuführen, die vollständig von AWS verwaltet werden. Mit Amazon FSx können Sie die Features und die Leistung gängiger Open-Source-Dateisysteme und kommerziell lizenzierter Dateisysteme nutzen, um zeitaufwändige Verwaltungsaufgaben zu vermeiden. Weitere Informationen finden Sie in der Amazon-FSx-Dokumentation.
- Amazon Simple Notification Service (SNS) ist ein vollständig verwalteter Messaging-Dienst für Kommunikation application-to-application und application-to-person Kommunikation. Sie können

Amazon SNS für die Überwachung durch Application Insights konfigurieren. Wenn Amazon SNS als Ressource für die Überwachung konfiguriert ist, verfolgt Application Insights SNS-Metriken, um festzustellen, warum SNS-Nachrichten Probleme haben oder fehlschlagen.

- Amazon Elastic File System (Amazon EFS) ist ein vollständig verwaltetes elastisches NFS-Dateisystem zur Verwendung mit AWS Cloud Services und lokalen Ressourcen. Es ist so konzipiert, dass es bei Bedarf auf Petabyte skaliert werden kann, ohne die Anwendungen zu unterbrechen. Es wird automatisch erweitert oder verkleinert, wenn Sie Dateien hinzufügen oder entfernen, wodurch die Notwendigkeit entfällt, Kapazität bereitzustellen und zu verwalten, um dem Wachstum gerecht zu werden. Weitere Informationen finden Sie in der Amazon-Elastic-File-System-Dokumentation.

Ähnliche Drittanbieterdienste

- Für einige Workloads und Anwendungen, die in Application Insights überwacht werden, wird der Prometheus JMX Exporter mithilfe von AWS Systems Manager Distributor installiert, sodass CloudWatch Application Insights Java-spezifische Metriken abrufen kann. Wenn Sie eine Java-Anwendung überwachen möchten, installiert Application Insights automatisch den Prometheus-JMX-Exporter für Sie.

Unterstützte Anwendungskomponenten

CloudWatch Application Insights scannt Ihre Ressourcengruppe, um Anwendungskomponenten zu identifizieren. Komponenten können eigenständig, automatisch gruppiert (z. B. Instances in einer Auto-Scaling-Gruppe oder hinter einem Load Balancer) oder benutzerdefiniert (durch Gruppierung einzelner Amazon-EC2-Instances) sein.

Die folgenden Komponenten werden von CloudWatch Application Insights unterstützt:

AWS Komponenten

- Amazon EC2
- Amazon EBS
- Amazon RDS
- Elastic Load Balancing: Application Load Balancer und Classic Load Balancer (alle Ziel-Instances dieser Load Balancer werden identifiziert und konfiguriert).
- Amazon EC2 Auto Scaling-Gruppen: AWS Auto Scaling (Auto Scaling Scaling-Gruppen werden dynamisch für alle Ziel-Instances konfiguriert; wenn Ihre Anwendung skaliert wird, konfiguriert

CloudWatch Application Insights die neuen Instances automatisch). Auto Scaling Scaling-Gruppen werden für CloudFormation stapelbasierte Ressourcengruppen nicht unterstützt.

- AWS Lambda
- Amazon-Simple-Queue-Service (Amazon SQS)
- Amazon-DynamoDB-Tabelle.
- Amazon-S3-Bucket-Metriken
- AWS Step Functions
- Amazon-API-Gateway-REST-API-Phasen
- Amazon Elastic Container Service (Amazon ECS): Cluster, Service und Aufgabe
- Amazon Elastic Kubernetes Service (Amazon EKS): Cluster
- Kubernetes auf Amazon EC2: Kubernetes-Cluster läuft auf EC2
- Amazon SNS-Thema

Ressourcen anderer Komponententypen werden derzeit nicht von CloudWatch Application Insights nachverfolgt. Wenn ein unterstützter Komponententyp nicht in Ihrer Application Insights-Anwendung erscheint, kann es sein, dass die Komponente bereits von einer anderen Anwendung registriert und verwaltet wird, die Sie besitzen und die von Application Insights überwacht wird.

Unterstützte Technologie-Stacks

Sie können CloudWatch Application Insights verwenden, um Ihre Anwendungen zu überwachen, die auf Windows Server- und Linux-Betriebssystemen ausgeführt werden, indem Sie die Dropdownmenüoption Anwendungsebene für eine der folgenden Technologien auswählen:

- Frontend: Microsoft Internet Information Services (IIS) Webserver
- Worker-Ebene:
  - .NET Framework.
  - .NET Core
- Anwendungen:
  - Java
  - NetWeaver SAP-Standardbereitstellungen, verteilte Bereitstellungen und Bereitstellungen mit hoher Verfügbarkeit
- Active Directory

- SharePoint
- Datenbanken:
  - Microsoft SQL Server in Amazon RDS oder Amazon EC2 (einschließlich SQL Server-Hochverfügbarkeitskonfigurationen. Siehe Beispiele für die Komponentenkonfiguration)
  - MySQL läuft auf Amazon RDS, Amazon Aurora oder Amazon EC2
  - PostgreSQL läuft auf Amazon RDS oder Amazon EC2
  - Amazon-DynamoDB-Tabelle.
  - Oracle läuft auf Amazon RDS oder Amazon EC2
  - SAP HANA-Datenbank auf einer einzelnen Amazon-EC2-Instance und mehreren EC2-Instances
  - AZ-übergreifende Einrichtung der SAP HANA-Datenbank mit hoher Verfügbarkeit
  - SAP Sybase ASE-Datenbank auf einer einzelnen Amazon EC2 EC2-Instance
  - AZ-übergreifende Einrichtung der SAP Sybase ASE-Datenbank mit hoher Verfügbarkeit

Wenn keiner der oben aufgeführten Technologiestacks auf Ihre Anwendungsressourcen zutrifft, können Sie Ihren Anwendungsstack überwachen, indem Sie im Dropdown-Menü der Anwendungsebene auf der Seite Manage monitoring (Überwachung verwalten) die Option Custom (Benutzerdefiniert) wählen.

So funktioniert Amazon CloudWatch Application Insights

Dieser Abschnitt enthält Informationen zur Funktionsweise von CloudWatch Application Insights, darunter:

- So überwacht Application Insights Anwendungen
- Datenaufbewahrung
- Kontingente
- AWS Von CloudWatch Application Insights verwendete Systems Manager (SSM) -Pakete
- AWS Von CloudWatch Application Insights verwendete Systems Manager (SSM) -Dokumente

So überwacht Application Insights Anwendungen

Application Insights überwacht Anwendungen wie folgt.

Anwendungserkennung und -konfiguration

Wenn eine Anwendung zum ersten Mal zu CloudWatch Application Insights hinzugefügt wird, werden die Anwendungskomponenten gescannt, um wichtige Kennzahlen, Protokolle und andere Datenquellen zur Überwachung Ihrer Anwendung zu empfehlen. Sie können Ihre Anwendung dann auf der Grundlage dieser Empfehlungen konfigurieren.

Datenvorverarbeitung

CloudWatch Application Insights analysiert kontinuierlich die Datenquellen, die in allen Anwendungsressourcen überwacht werden, um metrische Anomalien zu erkennen und Fehler (Beobachtungen) zu protokollieren.

Intelligente Problemerkennung

Die CloudWatch Application Insights-Engine erkennt Probleme in Ihrer Anwendung, indem sie Beobachtungen mithilfe von Klassifizierungsalgorithmen und integrierten Regeln korreliert. Zur Unterstützung bei der Fehlerbehebung erstellt sie automatisierte CloudWatch Dashboards, die kontextbezogene Informationen zu den Problemen enthalten.

Alarm und Aktion

Wenn CloudWatch Application Insights ein Problem mit Ihrer Anwendung erkennt, generiert es CloudWatch Ereignisse, um Sie über das Problem zu informieren. Weitere Informationen darüber, wie Sie diese Ereignisse einrichten, finden Sie unter Application Insights — CloudWatch Ereignisse und Benachrichtigungen bei erkannten Problemen.

Beispielszenario

Sie haben eine ASP.NET-Anwendung, die von einer SQL Server-Datenbank unterstützt wird. Plötzlich beginnt Ihre Datenbank aufgrund der hohen Speicherauslastung zu versagen. Dies führt zu einer Verschlechterung der Anwendungsleistung und möglicherweise zu HTTP-500-Fehlern auf den Webservern und im Load Balancer.

Mit CloudWatch Application Insights und seinen intelligenten Analysen können Sie die Anwendungsebene identifizieren, die das Problem verursacht, indem Sie das dynamisch erstellte Dashboard überprüfen, das die zugehörigen Metriken und Protokolldateiausschnitte anzeigt. In diesem Fall kann das Problem auf der SQL-Datenbankschicht liegen.

Datenaufbewahrung

CloudWatch Application Insights speichert Probleme 55 Tage und Beobachtungen 60 Tage lang.

Kontingente

Standardkontingente für CloudWatch Application Insights finden Sie unter Amazon CloudWatch Application Insights-Endpunkte und Kontingente. Sofern nicht anders angegeben, gilt jedes Kontingent pro AWS Region. Kontaktieren Sie den AWS Support, um eine Erhöhung Ihres Service-Kontingents zu beantragen. Viele Services enthalten Kontingente, die nicht geändert werden können. Weitere Informationen zu den Kontingenten für einen bestimmten Service finden Sie in der Dokumentation des jeweiligen Service.

AWS Von CloudWatch Application Insights verwendete Systems Manager (SSM) - Pakete

Die in diesem Abschnitt aufgeführten Pakete werden von Application Insights verwendet und können unabhängig voneinander mit AWS Systems Manager Distributor verwaltet und bereitgestellt werden. Weitere Informationen zum SSM-Distributor finden Sie unter AWS Systems Manager Distributor im AWS -Benutzerhandbuch zu Systems Manager.

Pakete:

- AWSObservabilityExporter-JMXExporterInstallAndConfigure
- AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure
- AWSObservabilityExporter-HAClusterExporterInstallAndConfigure
- AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure
- AWSObservabilityExporter-SQLExporterInstallAndConfigure

`AWSObservabilityExporter-JMXExporterInstallAndConfigure`

Sie können Workload-spezifische Java-Metriken vom Prometheus-JMX-Exporter für Application Insights abrufen, um Alarme zu konfigurieren und zu überwachen. Wählen Sie in der Application Insights-Konsole auf der Seite Überwachung verwalten die Option JAVA-Anwendung aus der Dropdown-Liste Anwendungsebene aus. Wählen Sie dann unter JAVA-Prometheus-Exporter-Konfiguration Ihre Sammlungsmethode und die JMX-Portnummer aus.

Gehen Sie wie folgt vor, um AWS Systems Manager Distributor zum Verpacken, Installieren und Konfigurieren des AWS bereitgestellten Prometheus JMX-Exportpakets unabhängig von Application Insights zu verwenden.

Voraussetzungen für die Verwendung des SSM-Pakets Prometheus JMX Exporter

- Installierte SSM-Agent Version 2.3.1550.0 oder höher
- Die Umgebungsvariable JAVA_HOME ist festgelegt

Installieren und konfigurieren Sie das **`AWSObservabilityExporter-JMXExporterInstallAndConfigure`**-Paket

Das `AWSObservabilityExporter-JMXExporterInstallAndConfigure`-Paket ist ein SSM-Distributor-Paket, mit dem Sie Prometheus-JMX Exporter installieren und konfigurieren können. Wenn Java-Metriken vom Prometheus JMX-Exporter gesendet werden, kann der CloudWatch Agent so konfiguriert werden, dass er die Metriken für den Service abruft. CloudWatch

1. Bereiten Sie entsprechend Ihren Einstellungen die YAML-Konfigurationsdatei für den Prometheus JMX-Exporter vor, die sich im Prometheus-Repository befindet. GitHub Orientieren Sie sich an den Beispielen für die Konfiguration und den Optionsbeschreibungen.
2. Kopieren Sie die als Base64 codierte YAML-Konfigurationsdatei des Prometheus-JMX-Exporters in einen neuen SSM-Parameter im SSM Parameter Store.
3. Navigieren Sie zur SSM-Distributor-Konsole und öffnen Sie die Registerkarte Owned by Amazon (Im Besitz von Amazon). Wählen Sie -JMX und wählen Sie einmal InstallierenAWSObservabilityExporter. ExporterInstallAndConfigure
4. Aktualisieren Sie den im ersten Schritt erstellten SSM-Parameter, indem Sie „Zusätzliche Argumente“ durch Folgendes ersetzen:

```
{
  "SSM_EXPORTER_CONFIGURATION": "{{ssm:<SSM_PARAMETER_STORE_NAME>}}",
  "SSM_EXPOSITION_PORT": "9404"
}
```

> **Note**
>
> Port 9404 ist der Standardport, der zum Senden von Prometheus-JMX-Metriken verwendet wird. Sie können diesen Port ändern.

Beispiel: Konfigurieren Sie den CloudWatch Agenten zum Abrufen von Java-Metriken

1. Installieren Sie den Prometheus-JMX-Exporter, wie im vorherigen Verfahren beschrieben. Überprüfen Sie dann, ob er korrekt auf Ihrer Instance installiert ist, indem Sie den Portstatus kontrollieren.

   Beispiel für erfolgreiche Installation auf Windows-Instance

   ```
   PS C:\> curl http://localhost:9404 (http://localhost:9404/)
   StatusCode : 200
   StatusDescription : OK
   Content : # HELP jvm_info JVM version info
   ```

   Beispiel für erfolgreiche Installation auf Linux-Instance

   ```
   $ curl localhost:9404
   # HELP jmx_config_reload_failure_total Number of times configuration have failed to
    be reloaded.
   # TYPE jmx_config_reload_failure_total counter
   jmx_config_reload_failure_total 0.0
   ```

2. Erstellen Sie die YAML-Datei für die Prometheus-Dienstermittlung. Das folgende Beispiel für die Dienstermittlungsdatei führt Folgendes aus:

   - Gibt den Prometheus-JMX-Exporter-Host-Port als `localhost: 9404` an.
   - Fügt den Metriken Beschriftungen (`ApplicationComponentName`, und`InstanceId`) hinzu, die als CloudWatch Metrikdimensionen festgelegt werden können.

   ```
   $ cat prometheus_sd_jmx.yaml
   - targets:
     - 127.0.0.1:9404
     labels:
       Application: myApp
       ComponentName: arn:aws:elasticloadbalancing:us-
   east-1:123456789012:loadbalancer/app/sampl-Appli-MMZW8E3GH4H2/aac36d7fea2a6e5b
       InstanceId: i-12345678901234567
   ```

3. Erstellen Sie die YAML-Konfigurationsdatei für den Prometheus-JMX-Exporter. Die folgende Beispielkonfigurationsdatei gibt Folgendes an:

- Das Intervall für das Abrufen von Metriken und der Zeitüberschreitungszeitraum.
- Die Metrikabrufaufträge (auch als „Scraping“ bezeichnet) (`jmx` und `sap`), die den Auftragsnamen, die maximal gleichzeitig zurückgegebene Zeitreihe und den Dateipfad zur Diensterkennung enthalten.

```
$ cat prometheus.yaml
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: jmx
    sample_limit: 10000
    file_sd_configs:
      - files: ["/tmp/prometheus_sd_jmx.yaml"]
  - job_name: sap
    sample_limit: 10000
    file_sd_configs:
      - files: ["/tmp/prometheus_sd_sap.yaml"]
```

4. Stellen Sie sicher, dass der CloudWatch Agent auf Ihrer Amazon EC2 EC2-Instance installiert ist und dass die Version 1.247346.1b249759 oder höher ist. Informationen zur Installation des Agenten auf Ihrer EC2-Instance finden Sie unter Installation des CloudWatch Agenten. CloudWatch Informationen zur Überprüfung der Version finden Sie unter Informationen zu CloudWatch Agentenversionen finden.
5. Konfigurieren Sie den CloudWatch Agenten. Weitere Informationen zur Konfiguration der CloudWatch Agenten-Konfigurationsdatei finden Sie unter Manuelles Erstellen oder Bearbeiten der CloudWatch Agenten-Konfigurationsdatei. Mit der folgenden Beispieldatei für die CloudWatch Agentenkonfiguration wird Folgendes ausgeführt:

   - Gibt den Konfigurationsdateipfad für den Prometheus-JMX-Exporter an.
   - Gibt die Zielprotokollgruppe an, in der EMF-Metrikprotokolle veröffentlicht werden sollen.
   - Gibt zwei Sätze von Dimensionen für jeden Metriknamen an.
   - Sendet 8 (4 Metriknamen x 2 Sätze von Dimensionen pro Metrikname) CloudWatch Metriken.

```
{
   "logs":{
      "logs_collected":{
```

```
        ....
    },
    "metrics_collected":{
       "prometheus":{
          "cluster_name":"prometheus-test-cluster",
          "log_group_name":"prometheus-test",
          "prometheus_config_path":"/tmp/prometheus.yaml",
          "emf_processor":{
             "metric_declaration_dedup":true,
             "metric_namespace":"CWAgent",
             "metric_unit":{
                "jvm_threads_current":"Count",
                "jvm_gc_collection_seconds_sum":"Second",
                "jvm_memory_bytes_used":"Bytes"
             },
             "metric_declaration":[
                {
                   "source_labels":[
                      "job"
                   ],
                   "label_matcher":"^jmx$",
                   "dimensions":[
                      [
                         "InstanceId",
                         "ComponentName"
                      ],
                      [
                         "ComponentName"
                      ]
                   ],
                   "metric_selectors":[
                      "^java_lang_threading_threadcount$",
                      "^java_lang_memory_heapmemoryusage_used$",
                      "^java_lang_memory_heapmemoryusage_committed$"
                   ]
                }
             ]
          }
       }
    }
 },
 "metrics":{
    ....
 }
```

```
}
```

`AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure`

Sie können Workload-spezifische SAP HANA-Metriken vom Prometheus-HANA-Datenbank-Exporter für Application Insights abrufen, um Alarme zu konfigurieren und zu überwachen. Weitere Informationen finden Sie unter Einrichten der SAP HANA-Datenbank für die Überwachung in diesem Handbuch.

Gehen Sie wie folgt vor, um AWS Systems Manager Distributor zum Verpacken, Installieren und Konfigurieren des AWS bereitgestellten Prometheus HANA-Datenbank-Exporter-Pakets unabhängig von Application Insights zu verwenden.

Voraussetzungen für die Verwendung des Prometheus-HANA-Datenbank-Exporter-SSM-Pakets

- Installierte SSM-Agent Version 2.3.1550.0 oder höher
- SAP HANA-Datenbank
- Linux-Betriebssystem (SUSE Linux, Linux) RedHat
- Ein Secret mit Anmeldeinformationen zur Überwachung der SAP HANA-Datenbank mit AWS Secrets Manager. Erstellen Sie ein Secret mit Schlüssel-/Wert-Paaren, geben Sie den Schlüsselbenutzernamen an und geben Sie den Datenbankbenutzer für den Wert ein. Fügen Sie ein zweites Schlüsselpasswort hinzu und geben Sie dann als Wert das Passwort ein. Weitere Informationen zur Erstellung von Secrets finden Sie unter Erstellen von Secrets im AWS Secrets Manager -Benutzerhandbuch. Das Secret muss wie folgt formatiert sein:

```
{
  "username": "<database_user>",
  "password": "<database_password>"
}
```

Installieren und konfigurieren Sie das **`AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure`**-Paket

Das Paket `AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure` ist ein SSM-Distributor-Paket, mit dem Sie den Prometheus-HANA-Datenbank-Exporter installieren und konfigurieren können. Wenn HANA-Datenbankmetriken vom Prometheus HANA-Datenbank-Exporter

gesendet werden, kann der CloudWatch Agent so konfiguriert werden, dass er die Metriken für den Service abruft. CloudWatch

1. Erstellen Sie einen SSM-Parameter im SSM-Parameterspeicher, um die Exporter-Konfigurationen zu speichern. Im Folgenden sehen Sie einen beispielhaften Parameterwert.

```
{\"exposition_port\":9668,\"multi_tenant\":true,\"timeout\":600,\"hana\":{\"host\":
\"localhost\",\"port\":30013,\"aws_secret_name\":\"HANA_DB_CREDS\",\"scale_out_mode
\":true}}
```

> **Note**
>
> In diesem Beispiel läuft der Export nur auf der Amazon-EC2-Instance mit der aktiven SYSTEM-Datenbank. Auf den anderen EC2-Instances läuft er hingegen nicht, um doppelte Metriken zu vermeiden. Der Exporter kann alle Informationen des Datenbankmandanten aus der SYSTEM-Datenbank ziehen.

2. Erstellen Sie einen SSM-Parameter im SSM Parameter Store (Parameterspeicher), um die Exporter-Metrikabfragen zu speichern. Das Paket kann mehr als einen Metrikparameter akzeptieren. Jeder Parameter muss ein gültiges JSON-Objektformat haben. Im Folgenden sehen Sie einen beispielhaften Parameterwert:

```
{\"SELECT MAX(TIMESTAMP) TIMESTAMP, HOST, MEASURED_ELEMENT_NAME CORE,
 SUM(MAP(CAPTION, 'User Time', TO_NUMBER(VALUE), 0)) USER_PCT, SUM(MAP(CAPTION,
 'System Time', TO_NUMBER(VALUE), 0)) SYSTEM_PCT, SUM(MAP(CAPTION, 'Wait
 Time', TO_NUMBER(VALUE), 0)) WAITIO_PCT, SUM(MAP(CAPTION, 'Idle Time', 0,
 TO_NUMBER(VALUE))) BUSY_PCT, SUM(MAP(CAPTION, 'Idle Time', TO_NUMBER(VALUE), 0))
 IDLE_PCT FROM sys.M_HOST_AGENT_METRICS WHERE MEASURED_ELEMENT_TYPE = 'Processor'
 GROUP BY HOST, MEASURED_ELEMENT_NAME;\":{\"enabled\":true,\"metrics\":[{\"name\":
\"hanadb_cpu_user\",\"description\":\"Percentage of CPU time spent by HANA DB in user
 space, over the last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE\"],\"value\":
\"USER_PCT\",\"unit\":\"percent\",\"type\":\"gauge\"},{\"name\":\"hanadb_cpu_system
\",\"description\":\"Percentage of CPU time spent by HANA DB in Kernel space,
 over the last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE\"],\"value\":
\"SYSTEM_PCT\",\"unit\":\"percent\",\"type\":\"gauge\"},{\"name\":\"hanadb_cpu_waitio
\",\"description\":\"Percentage of CPU time spent by HANA DB in IO mode, over the
 last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE\"],\"value\":\"WAITIO_PCT\",
\"unit\":\"percent\",\"type\":\"gauge\"},{\"name\":\"hanadb_cpu_busy\",\"description
\":\"Percentage of CPU time spent by HANA DB, over the last minute (in seconds)\",
\"labels\":[\"HOST\",\"CORE\"],\"value\":\"BUSY_PCT\",\"unit\":\"percent\",\"type\":
```

```
\"gauge\"},{\"name\":\"hanadb_cpu_idle\",\"description\":\"Percentage of CPU time not
 spent by HANA DB, over the last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE
\"],\"value\":\"IDLE_PCT\",\"unit\":\"percent\",\"type\":\"gauge\"}]}}
```

Weitere Informationen zu Metrikabfragen finden Sie im Repo unter SUSE / hanadb_exporter. GitHub

3. Navigieren Sie zur SSM-Distributor-Konsole und öffnen Sie die Registerkarte Owned by Amazon (Im Besitz von Amazon). Wählen Sie AWSObservabilityExporter-SAP-HANADB ExporterInstallAndConfigure * und wählen Sie einmal Installieren.
4. Aktualisieren Sie den im ersten Schritt erstellten SSM-Parameter, indem Sie „Zusätzliche Argumente“ durch Folgendes ersetzen:

```
{
  "SSM_EXPORTER_CONFIG": "{{ssm:<*SSM_CONFIGURATIONS_PARAMETER_STORE_NAME>*}}",
  "SSM_SID": "<SAP_DATABASE_SID>",
  "SSM_EXPORTER_METRICS_1": "{{ssm:<SSM_FIRST_METRICS_PARAMETER_STORE_NAME>}}",
  "SSM_EXPORTER_METRICS_2": "{{ssm:<SSM_SECOND_METRICS_PARAMETER_STORE_NAME>}}"
}
```

5. Wählen Sie die Amazon-EC2-Instances mit SAP HANA-Datenbank aus und dann Run (Ausführen).

`AWSObservabilityExporter-HAClusterExporterInstallAndConfigure`

Sie können Workload-spezifische Hochverfügbarkeits-Cluster-Metriken (High Availiablility, HA) aus dem Prometheus-HANA-Cluster-Exporter abrufen, um mit Application Insights Alarme für eine Einrichtung der SAP HANA-Datenbank mit hoher Verfügbarkeit zu konfigurieren. Weitere Informationen finden Sie unter Einrichten der SAP HANA-Datenbank für die Überwachung in diesem Handbuch.

Gehen Sie wie folgt vor, um AWS Systems Manager Distributor zum Verpacken, Installieren und Konfigurieren des AWS bereitgestellten Prometheus HA-Cluster-Exportpakets unabhängig von Application Insights zu verwenden.

Voraussetzungen für die Verwendung des Prometheus-HA-Cluster-Exporter-SSM-Pakets

- Installierte SSM-Agent Version 2.3.1550.0 oder höher
- HA-Cluster für Pacemaker, Corosync, SBD, and DRBD
- Linux-Betriebssystem (SUSE Linux, Linux) RedHat

Installieren und konfigurieren Sie das **AWSObservabilityExporter-HAClusterExporterInstallAndConfigure**-Paket

Das AWSObservabilityExporter-HAClusterExporterInstallAndConfigure-Paket ist ein SSM-Distributor-Paket, mit dem Sie Prometheus-HA Cluster Exporter installieren und konfigurieren können. Wenn Cluster-Metriken vom Prometheus HANA-Datenbank-Exporter gesendet werden, kann der CloudWatch Agent so konfiguriert werden, dass er die Metriken für den Service abruft. CloudWatch

1. Erstellen Sie einen SSM-Parameter im SSM Parameter Store (Parameterspeicher), um die Exporter-Konfigurationen im Format JSON zu speichern. Im Folgenden sehen Sie einen beispielhaften Parameterwert.

   ```
   {\"port\":\"9664\",\"address\":\"0.0.0.0\",\"log-level\":\"info\",\"crm-mon-path
   \":\"/usr/sbin/crm_mon\",\"cibadmin-path\":\"/usr/sbin/cibadmin\",\"corosync-
   cfgtoolpath-path\":\"/usr/sbin/corosync-cfgtool\",\"corosync-quorumtool-path\":\"/
   usr/sbin/corosync-quorumtool\",\"sbd-path\":\"/usr/sbin/sbd\",\"sbd-config-path\":
   \"/etc/sysconfig/sbd\",\"drbdsetup-path\":\"/sbin/drbdsetup\",\"enable-timestamps
   \":false}
   ```

   Weitere Informationen zu den Exporter-Konfigurationen finden Sie im Repo unter. ClusterLabs / ha_cluster_exporter GitHub
2. Navigieren Sie zur SSM-Distributor-Konsole und öffnen Sie die Registerkarte Owned by Amazon (Im Besitz von Amazon). Wählen Sie AWSObservabilityExporter-HA ClusterExporterInstallAndConfigure * und wählen Sie einmal Installieren.
3. Aktualisieren Sie den im ersten Schritt erstellten SSM-Parameter, indem Sie „Zusätzliche Argumente“ durch Folgendes ersetzen:

   ```
   {
     "SSM_EXPORTER_CONFIG": "{{ssm:<*SSM_CONFIGURATIONS_PARAMETER_STORE_NAME>*}}"
   }
   ```
4. Wählen Sie die Amazon-EC2-Instances mit SAP HANA-Datenbank aus und dann Run (Ausführen).

AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure

Sie können Workload-spezifische NetWeaver SAP-Metriken vom Prometheus SAP Host Exporter for Application Insights abrufen, um Alarme für SAP NetWeaver Distributed- und High Availability-

Bereitstellungen zu konfigurieren und zu überwachen. Weitere Informationen finden Sie unter Erste Schritte mit Amazon CloudWatch Application Insights.

Um den AWS Systems Manager Distributor zu verwenden, um das SAP-Host-Exporter-Paket unabhängig von Application Insights zu verpacken, zu installieren und zu konfigurieren, führen Sie die folgenden Schritte aus.

Voraussetzungen für die Verwendung des SSM-Pakets Prometheus-SAP-Host-Explorer

- Installierte SSM-Agent Version 2.3.1550.0 oder höher
- SAP-Anwendungsserver NetWeaver
- Linux-Betriebssystem (SUSE Linux, RedHat Linux)

Installieren und konfigurieren Sie das **`AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure`**-Paket

Das `AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure` Paket ist ein SSM Distributor-Paket, mit dem Sie den SAP NetWeaver Prometheus Metrics Exporter installieren und konfigurieren können. Wenn NetWeaver SAP-Metriken vom Prometheus-Exporter gesendet werden, kann der CloudWatch Agent so konfiguriert werden, dass er die Metriken für den Service abruft. CloudWatch

1. Erstellen Sie einen SSM-Parameter im SSM Parameter Store (Parameterspeicher), um die Exporter-Konfigurationen im Format JSON zu speichern. Im Folgenden sehen Sie einen beispielhaften Parameterwert.

   ```
   {\"address\":\"0.0.0.0\",\"port\":\"9680\",\"log-level\":\"info\",\"is-HA\":false}
   ```

   - address

     Die Zieladresse, an die Prometheus Metriken gesendet werden sollen. Der Standardwert ist `localhost`.

   - port

     Der Ziel-Port, an den Prometheus-Metriken gesendet werden sollen. Der Standardwert ist `9680`.

   - is-HA

`true`für NetWeaver SAP-Hochverfügbarkeitsbereitstellungen. Für alle anderen Bereitstellungen ist der Wert `false`.

2. Navigieren Sie zur SSM-Distributor-Konsole und öffnen Sie die Registerkarte Owned by Amazon (Im Besitz von Amazon). Wählen Sie AWSObservabilityExporter-SAP-SAP aus HostExporterInstallAndConfigure und wählen Sie Einmal installieren aus.
3. Aktualisieren Sie den im ersten Schritt erstellten SSM-Parameter, indem Sie „Zusätzliche Argumente“ durch Folgendes ersetzen:

```
{
  "SSM_EXPORTER_CONFIG": "{{ssm:<SSM_CONFIGURATIONS_PARAMETER_STORE_NAME>}}",
  "SSM_SID": "<SAP_DATABASE_SID>",
  "SSM_INSTANCES_NUM": "<instances_number seperated by comma>"
}
```

Beispiel

```
{
  "SSM_EXPORTER_CONFIG": "{{ssm:exporter_config_paramter}}",
  "SSM_INSTANCES_NUM": "11,12,10",
  "SSM_SID": "PR1"
}
```

4. Wählen Sie die Amazon EC2 EC2-Instances mit NetWeaver SAP-Anwendungen aus und klicken Sie auf Ausführen.

> **Note**
>
> Der Prometheus-Exporter verarbeitet die NetWeaver SAP-Metriken auf einem lokalen Endpunkt. Auf den lokalen Endpunkt können nur die Betriebssystembenutzer der Amazon-EC2-Instance zugreifen. Deshalb stehen die Metriken nach der Installation des Exporter-Pakets allen Betriebssystembenutzern zur Verfügung. Der lokale Standardendpunkt ist `localhost:9680/metrics`.

`AWSObservabilityExporter-SQLExporterInstallAndConfigure`

Sie können Workload-spezifische SQL Server-Metriken vom Prometheus-SQL-Exporter für Application Insights abrufen, um wichtige Metriken zu überwachen.

Wenn Sie AWS Systems Manager -Distributor verwenden möchten, um das SQL-Exporter-Paket unabhängig von Application Insights zu verpacken, zu installieren und zu konfigurieren, führen Sie die folgenden Schritte aus.

Voraussetzungen für die Verwendung des SSM-Pakets Prometheus SQL Exporter

- Installierte SSM-Agent Version 2.3.1550.0 oder höher
- Amazon-EC2-Instance mit SQL Server auf Windows und aktivierter SQL-Server-Benutzerauthentifizierung.
- Ein SQL-Server-Benutzer mit den folgenden Berechtigungen:

```
GRANT VIEW ANY DEFINITION TO
```

```
GRANT VIEW SERVER STATE TO
```

- Ein Geheimnis, das die Datenbankverbindungszeichenfolge mit AWS Secrets Manager enthält. Weitere Informationen zur Erstellung von Secrets finden Sie unter Erstellen von Secrets im AWS Secrets Manager -Benutzerhandbuch. Das Secret muss wie folgt formatiert sein:

```
{
  "data_source_name":"sqlserver://<username>:<password>@localhost:1433"
}
```

> **Note**
>
> Wenn das Passwort oder der Benutzername Sonderzeichen enthält, müssen Sie die Sonderzeichen prozentual verschlüsseln, um eine erfolgreiche Verbindung zur Datenbank zu gewährleisten.

Installieren und konfigurieren Sie das **`AWSObservabilityExporter-SQLExporterInstallAndConfigure`**-Paket

Das `AWSObservabilityExporter-SQLExporterInstallAndConfigure`-Paket ist ein SSM-Distributor-Paket, mit dem Sie den Metrik-Exporter SQL Prometheus installieren und konfigurieren können. Wenn Metriken vom Prometheus-Exporter gesendet werden, kann der CloudWatch Agent so konfiguriert werden, dass er die Metriken für den Service abruft. CloudWatch

1. Bereiten Sie die SQL Exporter YAML-Konfiguration nach Ihren Vorstellungen vor. In der folgenden Beispielkonfiguration ist eine einzelne Metrik konfiguriert. Verwenden Sie die Beispielkonfiguration, um die Konfiguration mit zusätzlichen Metriken zu aktualisieren, oder erstellen Sie Ihre eigene Konfiguration.

```
---
global:
  scrape_timeout_offset: 500ms
  min_interval: 0s
  max_connections: 3
  max_idle_connections: 3
target:
  aws_secret_name: <SECRET_NAME>
  collectors:
    - mssql_standard
collectors:
  - collector_name: mssql_standard
    metrics:
    - metric_name: mssql_batch_requests
      type: counter
      help: 'Number of command batches received.'
      values: [cntr_value]
      query: |
        SELECT cntr_value
        FROM sys.dm_os_performance_counters WITH (NOLOCK)
        WHERE counter_name = 'Batch Requests/sec'
```

2. Kopieren Sie die als Base64 codierte YAML-Konfigurationsdatei des Prometheus-SQL-Exporters in einen neuen SSM-Parameter im SSM Parameter Store.
3. Navigieren Sie zur SSM-Distributor-Konsole und öffnen Sie die Registerkarte Owned by Amazon (Im Besitz von Amazon). Wählen Sie AWSObservabilityExporter-SQL ExporterInstallAndConfigure und anschließend „Einmal installieren“.
4. Ersetzen Sie die „Zusätzlichen Argumente“ durch die folgenden Informationen. Der `SSM_PARAMETER_NAME` ist der Name des Parameters, den Sie in Schritt 2 erstellt haben.

```
{
  "SSM_EXPORTER_CONFIGURATION":
    "{{ssm:<SSM_PARAMETER_STORE_NAME>}}",
    "SSM_PROMETHEUS_PORT": "9399",
    "SSM_WORKLOAD_NAME": "SQL"
}
```

5. Wählen Sie die Amazon-EC2-Instance mit der SQL-Server-Datenbank aus und klicken Sie dann auf Ausführen.

AWS Von CloudWatch Application Insights verwendete Systems Manager (SSM) - Dokumente

Application Insights verwendet die in diesem Abschnitt aufgeführten SSM-Dokumente, um die Aktionen zu definieren, die AWS Systems Manager auf Ihren verwalteten Instances ausführt. In diesen Dokumenten werden die `Run Command`-Funktionen von Systems Manager genutzt, um die Aufgaben zu automatisieren, die für die Ausführung der Überwachungsfunktionen von Application Insights erforderlich sind. Die Ausführungspläne für diese Dokumente werden von Application Insights verwaltet und können nicht geändert werden.

Weitere Informationen zu SSM-Dokumenten finden Sie unter AWS Systems Manager -Dokumente im Benutzerhandbuch für AWS Systems Manager .

Von CloudWatch Application Insights verwaltete Dokumente

In der folgenden Tabelle sind die SSM-Dokumente aufgeführt, die von Application Insights verwaltet werden.

Dokumentname	Beschreibung	Ausführungs-Zeitplan
`AWSEC2-DetectWorkload`	Erkennt automatisch Anwendungen, die in Ihrer Anwendungsumgebung ausgeführt werden und für die Überwachung durch Application Insights eingerichtet werden können.	Dieses Dokument wird stündlich in Ihrer Anwendungsumgebung ausgeführt, um up-to-date Anwendungsdetails abzurufen.

Dokumentname	Beschreibung	Ausführungs-Zeitplan
`AWSEC2-CheckPerformanceCounterSets`	Überprüft, ob Performance-Counter-Namespaces auf Ihren Amazon-EC2-Windows-Instances aktiviert sind.	Dieses Dokument wird stündlich in Ihrer Anwendungsumgebung ausgeführt und überwacht Performance-Counter-Metriken nur, wenn die entsprechenden Namespaces aktiviert sind.
`AWSEC2-ApplicationInsightsCloudwatchAgentInstallAndConfigure`	Installiert und konfiguriert den CloudWatch Agenten auf der Grundlage der Überwachungskonfiguration Ihrer Anwendungskomponenten.	Dieses Dokument wird alle 30 Minuten ausgeführt, um sicherzustellen, dass die CloudWatch Agentenkonfiguration immer korrekt ist und up-to-date. Das Dokument wird auch sofort ausgeführt, nachdem eine Änderung an der Einrichtung Ihrer Anwendungs-Überwachung vorgenommen wurde, z. B. das Hinzufügen oder Entfernen von Metriken oder das Aktualisieren von Protokollkonfigurationen.

Dokumente, die verwaltet werden von AWS Systems Manager

Die folgenden Dokumente werden von CloudWatch Application Insights verwendet und von Systems Manager verwaltet.

`AWS-ConfigureAWSPackage`

Application Insights verwendet dieses Dokument zur Installation und Deinstallation von Prometheus Exporter-Distributionspaketen, zur Erfassung von Workload-spezifischen Metriken und zur umfassenden Überwachung der Workloads auf Amazon EC2 EC2-Kundeninstanzen. CloudWatch

Application Insights installiert die Prometheus-Exportverteilerpakete nur, wenn der korrelierte Ziel-Workload auf Ihrer Instance ausgeführt wird.

In der folgenden Tabelle sind die Prometheus-Exporter-Verteilerpakete und die entsprechenden Ziel-Workloads aufgeführt.

Paketname des Prometheus-Exporter-Verteilers	Ziel-Workload
`AWSObservabilityExporter-HAClusterExporterInstallAndConfigure`	SAP HANA HA
`AWSObservabilityExporter-JMXExporterInstallAndConfigure`	Java/JMX
`AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure`	SAP HANA
`AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure`	NetWeaver
`AWSObservabilityExporter-SQLExporterInstallAndConfigure`	SQL Server (Windows) und SAP ASE (Linux)

`AmazonCloudWatch-ManageAgent`

Application Insights verwendet dieses Dokument, um den Status und die Konfiguration von CloudWatch Agent auf Ihren Instances zu verwalten und interne Metriken und Protokolle auf Systemebene von Amazon EC2 EC2-Instances betriebssystemübergreifend zu sammeln.

Erste Schritte mit Amazon CloudWatch Application Insights

Um mit CloudWatch Application Insights zu beginnen, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen und eine IAM-Richtlinie erstellt haben. Anschließend können Sie über den Konsolenlink loslegen, um CloudWatch Application Insights zu aktivieren. Um Ihre

Anwendungsressourcen zu konfigurieren, führen Sie die Schritte unter Einrichten, Konfigurieren und Verwalten der Anwendung für die Überwachung aus.

Inhalt

Rufen Sie CloudWatch Application Insights auf

Sie können über eine der folgenden Schnittstellen auf CloudWatch Application Insights zugreifen und diese verwalten:

- CloudWatch Konsole. Um Monitore für Ihre Anwendung hinzuzufügen, wählen Sie im linken Navigationsbereich der CloudWatch Konsole unter Insights die Option Application Insights aus. Nachdem Ihre Anwendung konfiguriert ist, können Sie die CloudWatch Konsole verwenden, um erkannte Probleme anzuzeigen und zu analysieren.
- AWS Befehlszeilenschnittstelle (AWS CLI). Sie können den verwenden AWS CLI , um auf AWS API-Operationen zuzugreifen. Weitere Informationen finden Sie unter Installation der AWS Befehlszeilenschnittstelle im Benutzerhandbuch für die AWS Befehlszeilenschnittstelle. Informationen zur Application Insights API finden Sie in der Amazon CloudWatch Application Insights API-Referenz.

Voraussetzungen

Sie müssen die folgenden Voraussetzungen erfüllen, um eine Anwendung mit CloudWatch Application Insights zu konfigurieren:

- AWS Systems Manager Aktivierung — Installieren Sie Systems Manager Agent (SSM Agent) auf Ihren Amazon EC2 EC2-Instances und aktivieren Sie die Instances für SSM. Informationen zur Installation des SSM-Agent finden Sie unter Einrichten von AWS Systems Manager im AWS Systems Manager -Benutzerhandbuch.
- EC2-Instance-Rolle – Sie müssen die folgenden Amazon-EC2-Instance-Rollen anhängen, um Systems Manager zu aktivieren.

  - Sie müssen die `AmazonSSMManagedInstanceCore`-Rolle anfügen, um Systems Manager zu aktivieren. Weitere Informationen finden Sie unter Beispiele für identitätsbasierte AWS Systems Manager -Richtlinien.
  - Sie müssen die `CloudWatchAgentServerPolicy` Richtlinie anhängen, damit Instance-Metriken und -Protokolle ausgegeben werden können. CloudWatch Weitere Informationen finden Sie unter IAM-Rollen und -Benutzer für die Verwendung mit CloudWatch Agenten erstellen.
- AWS Ressourcengruppen — Um Ihre Anwendungen in CloudWatch Application Insights zu integrieren, erstellen Sie eine Ressourcengruppe, die alle zugehörigen AWS Ressourcen umfasst, die von Ihrem Anwendungsstapel verwendet werden. Dazu gehören Anwendungs-Load-Balancer, Amazon-EC2-Instances, auf denen IIS und Web-Frontends ausgeführt werden, .NET Worker-Ebenen und SQL-Server-Datenbanken. Weitere Informationen zu Anwendungskomponenten und Technologie-Stacks, die von Application Insights unterstützt werden, finden Sie unterUnterstützte Anwendungskomponenten. CloudWatch Application Insights schließt automatisch Auto Scaling Scaling-Gruppen ein, die dieselben Tags oder CloudFormation Stacks wie Ihre Ressourcengruppe verwenden, da Auto Scaling Scaling-Gruppen von CloudFormation Ressourcengruppen nicht unterstützt werden. Weitere Informationen finden Sie unter Erste Schritte mit AWS Resource Groups.
- IAM-Berechtigungen — Für Benutzer, die keinen Administratorzugriff haben, müssen Sie eine AWS Identity and Access Management (IAM-) Richtlinie erstellen, die es Application Insights ermöglicht, eine dienstbezogene Rolle zu erstellen und sie der Benutzeridentität zuzuordnen. Informationen dazu, wie Sie eine IAM-Richtlinie erstellen, finden Sie unter IAM-Richtlinie.
- Dienstverknüpfte Rolle — Application Insights verwendet AWS Identity and Access Management (IAM) dienstbezogene Rollen. Es wird für Sie eine serviceverknüpfte Rolle erstellt, wenn Sie Ihre erste Application-Insights-Anwendung in der Application-Insights-Konsole erstellen. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für CloudWatch Application Insights.
- Unterstützung von Leistungszählermetriken für EC2-Windows-Instances: – Um Leistungszählermetriken auf Ihren Amazon-EC2-Windows-Instances zu überwachen, müssen Leistungszähler auf den Instances installiert sein. Informationen zu Performance Counter-Metriken und entsprechenden Performance Counter-Set-Namen finden Sie unter Performance Counter metrics. Weitere Informationen zu Leistungszählern finden Sie unter Leistungszähler.
- CloudWatch Amazon-Agent — Application Insights installiert und konfiguriert den CloudWatch Agenten. Wenn Sie den CloudWatch Agenten installiert haben, behält Application Insights Ihre Konfiguration bei. Um einen Zusammenführungskonflikt zu vermeiden, entfernen Sie die Konfiguration der Ressourcen, die Sie in Application Insights verwenden möchten, aus der

vorhandenen CloudWatch Agentenkonfigurationsdatei. Weitere Informationen finden Sie unter Erstellen oder bearbeiten Sie die CloudWatch Agenten-Konfigurationsdatei manuell.

IAM-Richtlinie

Um CloudWatch Application Insights verwenden zu können, müssen Sie eine AWS Identity and Access Management (IAM-) Richtlinie erstellen und sie Ihrem Benutzer, Ihrer Gruppe oder Rolle zuordnen. Weitere Informationen über Benutzer, Gruppen und Rollen finden Sie unter IAM-Identitäten (Benutzer, Benutzergruppen und Rollen). Die IAM-Richtlinie definiert die Benutzerberechtigungen.

Eine IAM-Richtlinie mithilfe der Konsole erstellen

Gehen Sie wie folgt vor, um eine IAM-Richtlinie mithilfe der IAM-Konsole zu erstellen.

1. Rufen Sie die IAM-Konsole auf Wählen Sie im linken Navigationsbereich Policies (Richtlinien).
2. Wählen Sie oben auf der Seite Create policy (Richtlinie erstellen).
3. Wählen Sie die Registerkarte JSON.
4. Kopieren und fügen Sie das folgende JSON-Dokument unter der Registerkarte JSON ein.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "applicationinsights:*",
                "iam:CreateServiceLinkedRole",
                "iam:ListRoles",
                "resource-groups:ListGroups"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

5. Wählen Sie Review policy (Richtlinie überprüfen).
6. Geben Sie einen Namen für die Richtlinie ein, zum Beispiel „AppInsightsPolicy.“ Geben Sie optional eine Description (Beschreibung) ein.
7. Wählen Sie Create Policy (Richtlinie erstellen).

8. Wählen Sie im linken Navigationsbereich Gruppen, Benutzer oder Rollen.
9. Wählen Sie den Namen der Benutzergruppe, des Benutzers oder der Rolle, an die Sie die Richtlinie anhängen möchten.
10. Wählen Sie Add permissions (Berechtigungen hinzufügen).
11. Wählen Sie die Option Attach existing policies directly (Vorhandene Richtlinien direkt anfügen) aus.
12. Suchen Sie nach der Richtlinie, die Sie gerade erstellt haben, und aktivieren Sie das Kontrollkästchen links neben dem Richtliniennamen.
13. Wählen Sie Next: Review (Weiter: Überprüfung).
14. Stellen Sie sicher, dass die richtige Richtlinie aufgelistet ist, und wählen Sie Add permissions (Berechtigungen hinzufügen).
15. Stellen Sie sicher, dass Sie sich mit dem Benutzer anmelden, der der Richtlinie zugeordnet ist, die Sie gerade erstellt haben, wenn Sie CloudWatch Application Insights verwenden.

Um eine IAM-Richtlinie zu erstellen, verwenden Sie AWS CLI

Um eine IAM-Richtlinie mit dem zu erstellen AWS CLI, führen Sie den Vorgang create-policy über die Befehlszeile aus und verwenden Sie dabei das obige JSON-Dokument als Datei in Ihrem aktuellen Ordner.

Um eine IAM-Richtlinie zu erstellen, verwenden Sie AWS Tools for Windows PowerShell

Um eine IAM-Richtlinie mit dem zu erstellen AWS Tools for Windows PowerShell, führen Sie das Cmdlt New-IamPolicy aus und verwenden Sie dabei das obige JSON-Dokument als Datei in Ihrem aktuellen Ordner.

IAM-Rollenberechtigungen für kontobasiertes Onboarding von Anwendungen

Wenn Sie alle Ressourcen in Ihrem Konto einbinden möchten und sich gegen die Von Application Insights verwaltete Richtlinie für vollen Zugriff auf die Funktionen von Application Insights entscheiden, müssen Sie Ihrer IAM-Rolle die folgenden Berechtigungen anfügen, damit Application Insights alle Ressourcen in Ihrem Konto erkennen kann:

```
"ec2:DescribeInstances"
"ec2:DescribeNatGateways"
"ec2:DescribeVolumes"
"ec2:DescribeVPCs"
```

```
"rds:DescribeDBInstances"
"rds:DescribeDBClusters"
"sqs:ListQueues"
"elasticloadbalancing:DescribeLoadBalancers"
"autoscaling:DescribeAutoScalingGroups"
"lambda:ListFunctions"
"dynamodb:ListTables"
"s3:ListAllMyBuckets"
"sns:ListTopics"
"states:ListStateMachines"
"apigateway:GET"
"ecs:ListClusters"
"ecs:DescribeTaskDefinition"
"ecs:ListServices"
"ecs:ListTasks"
"eks:ListClusters"
"eks:ListNodegroups"
"fsx:DescribeFileSystems"
"route53:ListHealthChecks"
"route53:ListHostedZones"
"route53:ListQueryLoggingConfigs"
"route53resolver:ListFirewallRuleGroups"
"route53resolver:ListFirewallRuleGroupAssociations"
"route53resolver:ListResolverEndpoints"
"route53resolver:ListResolverQueryLogConfigs"
"route53resolver:ListResolverQueryLogConfigAssociations"
"logs:DescribeLogGroups"
"resource-explorer:ListResources"
```

Einrichten, Konfigurieren und Verwalten der Anwendung für die Überwachung

Dieser Abschnitt enthält Schritte zum Einrichten, Konfigurieren und Verwalten Ihrer CloudWatch Application Insights-Anwendung mithilfe der Konsole, der und. AWS CLI AWS Tools for Windows PowerShell

Themen

- Richten Sie Ihre Anwendung für die Überwachung von der CloudWatch Konsole aus ein, konfigurieren und verwalten
- Richten Sie Ihre Anwendung für die Überwachung über die Befehlszeile ein, konfigurieren und verwalten Sie sie
- Application Insights — CloudWatch Ereignisse und Benachrichtigungen bei erkannten Problemen

Richten Sie Ihre Anwendung für die Überwachung von der CloudWatch Konsole aus ein, konfigurieren und verwalten

Dieser Abschnitt enthält Schritte zur Einrichtung, Konfiguration und Verwaltung Ihrer Anwendung für die Überwachung von der CloudWatch Konsole aus.

Konsolen-Verfahren

- Hinzufügen und Konfigurieren einer Anwendung
- Aktivieren von Application Insights zur Ressourcenüberwachung für Amazon ECS und Amazon EKS
- Deaktivieren der Überwachung einer Anwendungskomponente
- Löschen einer Anwendung

Hinzufügen und Konfigurieren einer Anwendung

Fügen Sie eine Anwendung von der CloudWatch Konsole aus hinzu und konfigurieren Sie sie

Gehen Sie wie folgt vor, um mit CloudWatch Application Insights von der CloudWatch Konsole aus zu beginnen.

1. Starten. Öffnen Sie die Landingpage der CloudWatch Konsole. Wählen Sie im linken Navigationsbereich unter Insights die Option Application Insights aus. Die sich öffnende Seite zeigt die Liste der Anwendungen, die mit CloudWatch Application Insights überwacht werden, zusammen mit ihrem Überwachungsstatus.
2. Hinzufügen einer Anwendung Um die Überwachung für Ihre Anwendung einzurichten, wählen Sie Add an application (Eine Anwendung hinzufügen) aus. Beim Auswählen der Option Add an application (Eine Anwendung hinzufügen) werden Sie aufgefordert, den Anwendungstyp auszuwählen (Choose Application Type).

    - Ressourcengruppenbasierte Anwendung. Mit dieser Option können Sie auswählen, welche Ressourcengruppen in diesem Konto überwacht werden sollen. Um mehrere Anwendungen auf einer Komponente zu verwenden, müssen Sie die ressourcengruppenbasierte Überwachung verwenden.
    - Kontobasierte Anwendung. Mit dieser Option können Sie alle Ressourcen in diesem Konto überwachen. Wenn Sie alle Ressourcen in einem Konto überwachen möchten, empfehlen wir diese Option eher als die ressourcengruppenbasierte Option, da der Onboarding-Prozess per Anwendung schneller ist.

> **Note**
>
> Sie können die ressourcengruppenbasierte Überwachung nicht mit kontobasierter Überwachung mithilfe von Application Insights kombinieren. Um den Anwendungstyp zu ändern, löschen Sie alle überwachten Anwendungen und wählen Sie Choose Application Type (Anwendungstyp auswählen) aus.

Wenn Sie Ihre erste Anwendung zur Überwachung hinzufügen, erstellt CloudWatch Application Insights eine dienstbezogene Rolle in Ihrem Konto, wodurch Application Insights berechtigt ist, andere AWS Dienste in Ihrem Namen aufzurufen. Weitere Informationen zu der serviceverknüpften Rolle, die von Application Insights in Ihrem Konto erstellt wurde, finden Sie unter Verwenden von serviceverknüpften Rollen für CloudWatch Application Insights.

3. Resource-based application monitoring

    1. Auswählen einer RessourcengruppeWählen Sie auf der Seite „Anwendungsdetails angeben“ aus der Dropdownliste die AWS Ressourcengruppe aus, die Ihre Anwendungsressourcen enthält. Zu diesen Ressourcen gehören Frontend-Server, Load Balancer, Auto Scaling-Gruppen und Datenbankserver.

        Wenn Sie keine Ressourcengruppe für Ihre Anwendung erstellt haben, können Sie das mit Create new resource group (Neue Ressourcengruppe erstellen) tun. Weitere Informationen zu Ressourcengruppen finden Sie im AWS Resource Groups -Benutzerhandbuch.

    2. CloudWatchEreignisse überwachen. Aktivieren Sie das Kontrollkästchen, um die Überwachung von Application Insights in CloudWatch Events zu integrieren und Einblicke aus Amazon EBS, Amazon EC2, Amazon ECS AWS CodeDeploy, AWS Health APIs And Notifications, Amazon RDS, Amazon S3 und zu erhalten. AWS Step Functions

    3. Integrieren Sie mit AWS Systems Manager OpsCenter. Aktivieren Sie das Kontrollkästchen Systems Manager für Abhilfemaßnahmen generieren, um Probleme anzuzeigen und sich benachrichtigen zu lassen, wenn Probleme OpsCenter OpsItems für ausgewählte Anwendungen erkannt werden. Geben Sie das SNS-Thema ARN an, um die Vorgänge nachzuverfolgen, die zur Lösung betrieblicher Arbeitsaufgaben (OpsItems) durchgeführt werden, die sich auf Ihre AWS Ressourcen beziehen.

4. Stichwörter — optional. CloudWatch Application Insights unterstützt sowohl tagbasierte als auch CloudFormation basierte Ressourcengruppen (mit Ausnahme von Auto Scaling Scaling-Gruppen). Weitere Informationen finden Sie unter Arbeiten mit dem Tag Editor.
5. Wählen Sie Weiter aus.

   Für die Anwendung wird ein ARN im folgenden Format generiert.

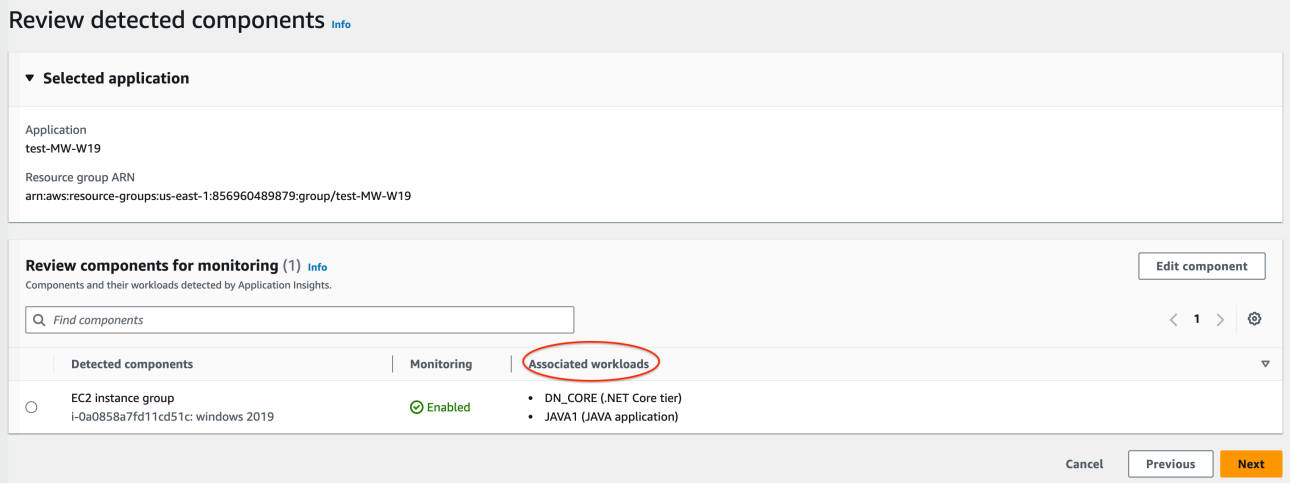
   ```
   arn:partition:applicationinsights:region:account-id:application/resource-group/resource-group-name
   ```

   Beispiel

   ```
   arn:aws:applicationinsights:us-east-1:123456789012:application/resource-group/my-resource-group
   ```

6. Auf der Seite Erkannte Komponenten überprüfen finden Sie unter Komponenten für die Überwachung überprüfen eine Tabelle mit den erkannten Komponenten und den zugehörigen erkannten Workloads.

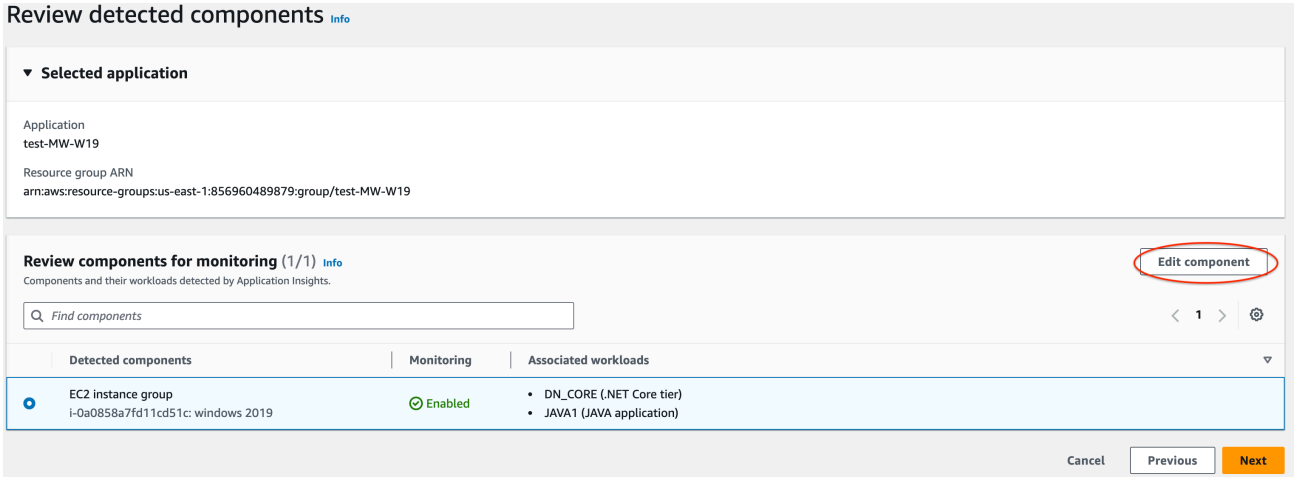
> **Note**
>
> Bei Komponenten, die mehrere benutzerdefinierte Workloads unterstützen, können Sie bis zu fünf Workloads für jede Komponente überwachen. Diese Workloads werden getrennt von der Komponente überwacht.



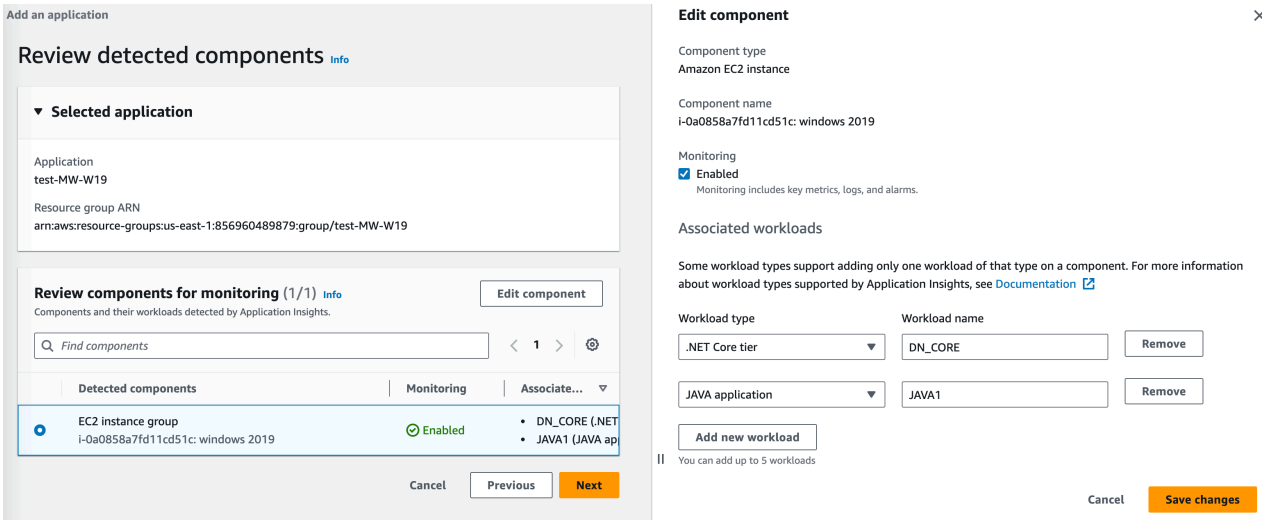
Unter Zugeordnete Workloads gibt es mehrere mögliche Meldungen, die angezeigt werden, wenn ein Workload nicht aufgeführt ist.

- Workloads konnten nicht erkannt werden – Beim Versuch, Workloads zu erkennen, ist ein Problem aufgetreten. Stellen Sie sicher, dass Sie Voraussetzungen abgeschlossen haben. Wenn Sie Workloads hinzufügen müssen, wählen Sie Komponente bearbeiten.
- Keine Workloads erkannt – Wir haben keine Workloads erkannt. Möglicherweise müssen Sie Workloads hinzufügen. Wählen Sie dazu Komponente bearbeiten.
- Nicht zutreffend – Die Komponente unterstützt keine benutzerdefinierten Workloads und wird mit Standardmetriken, Alarmen und Protokollen überwacht. Sie können diesen Komponenten keine Workloads hinzufügen.

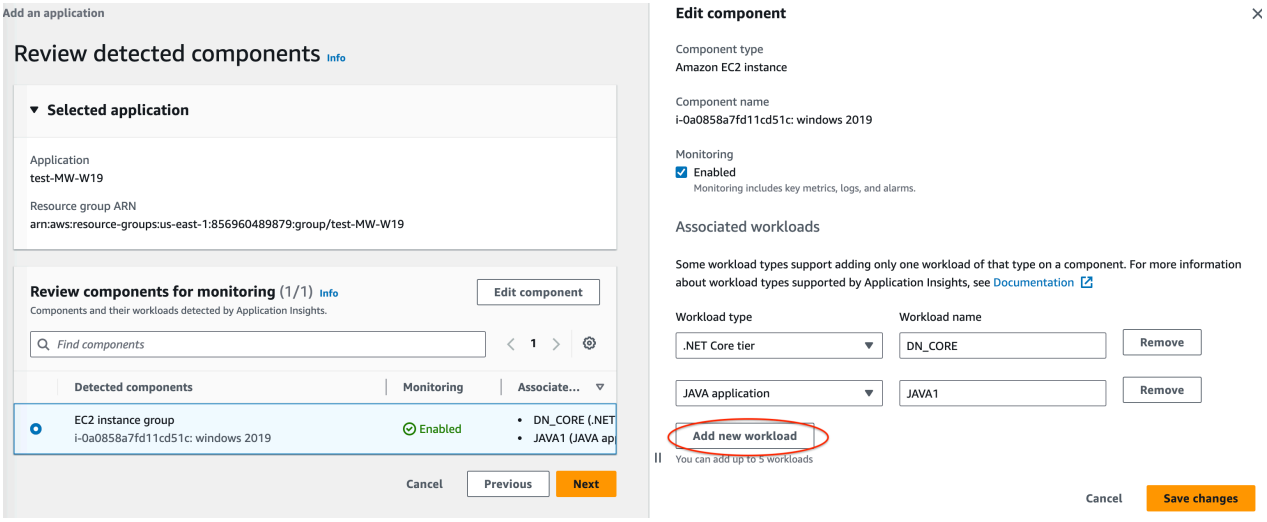
7. Um eine Komponente zu bearbeiten, wählen Sie eine Komponente aus und klicken Sie dann auf Komponente bearbeiten. Ein Seitenbereich wird geöffnet, in dem Workloads für die Komponente erkannt wurden. In diesem Bereich können Sie die Komponentendetails bearbeiten und neue Workloads hinzufügen.



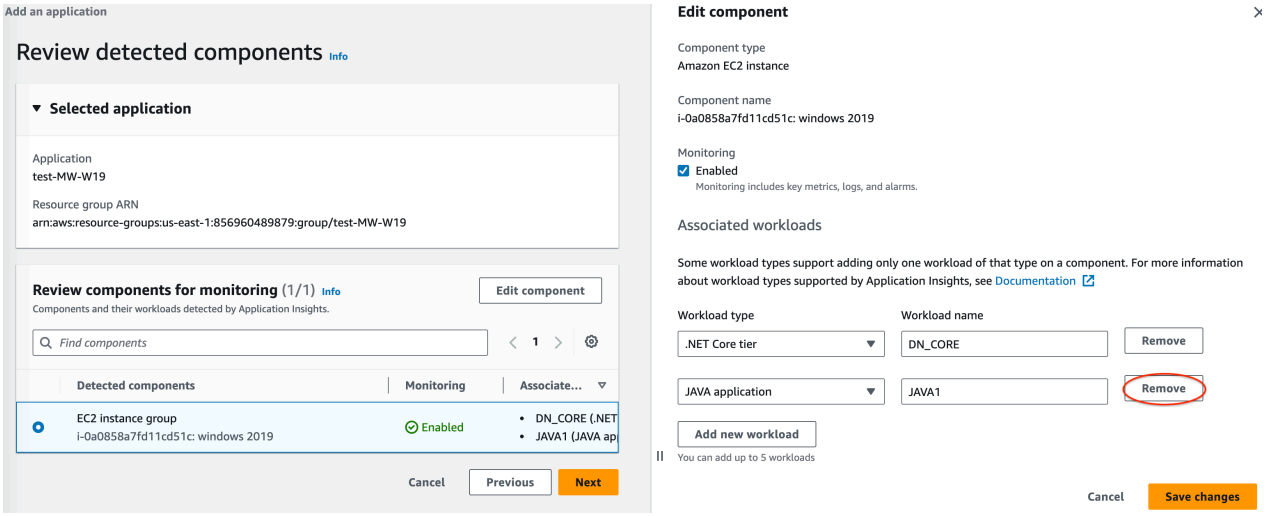
- Verwenden Sie die Dropdownliste, um den Workload-Typ oder -Namen zu bearbeiten.



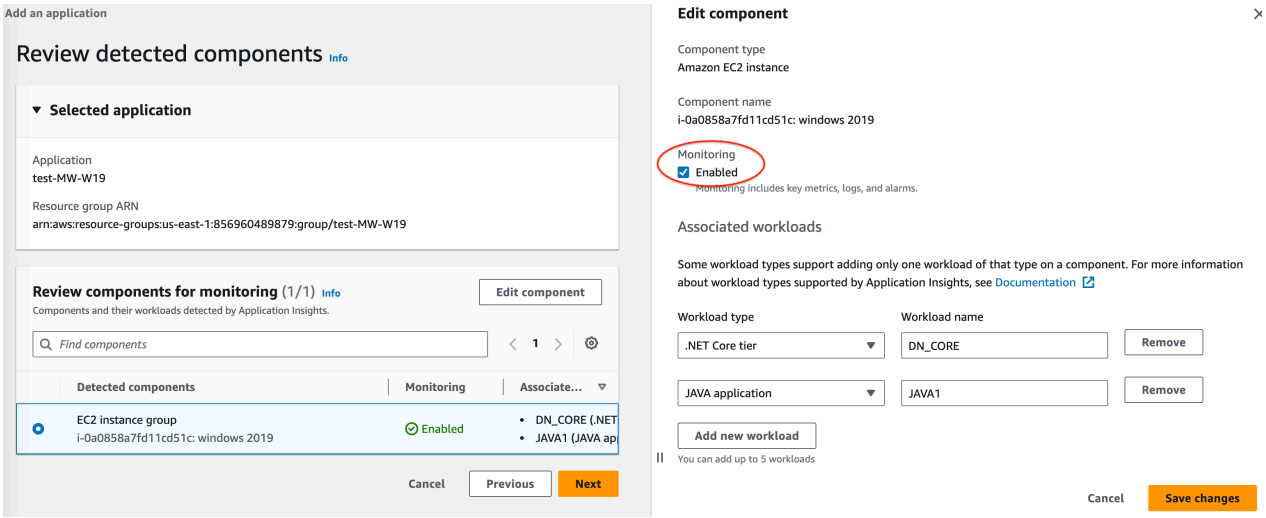
- Um der Komponente einen Workload hinzuzufügen, wählen Sie Neuen Workload hinzufügen.



  - Wenn Neuen Workload hinzufügen nicht angezeigt wird, unterstützt diese Komponente nicht mehrere Workloads.
  - Wenn die Überschrift Zugeordnete Workloads nicht angezeigt wird, unterstützt diese Komponente keine benutzerdefinierten Workloads.
- Um einen Workload zu entfernen, wählen Sie Entfernen neben dem Workload, den Sie aus der Überwachung entfernen möchten.



- Um die Überwachung für die gesamte Komponente zu deaktivieren, deaktivieren Sie das Kontrollkästchen Überwachen.



- Wenn Sie mit der Bearbeitung der Komponente fertig sind, wählen Sie in der unteren rechten Ecke die Option Änderungen speichern aus. Alle Änderungen an den Workloads für eine Komponente werden in der Tabelle Komponenten zur Überwachung überprüfen unter Zugeordnete Workloads angezeigt.

8. Wählen Sie auf der Seite Erkannte Komponenten überprüfen die Option Weiter.
9. Die Seite Komponentendetails angeben enthält alle Komponenten mit anpassbaren zugehörigen Workloads aus dem vorherigen Schritt.

> **Note**
>
> Wenn eine Komponentenüberschrift das Tag optional hat, sind zusätzliche Details für die Workloads in dieser Komponente optional.

Wenn eine Komponente auf dieser Seite nicht erscheint, verfügt die Komponente über keine zusätzlichen Details, die in diesem Schritt angegeben werden können.

10. Wählen Sie Weiter aus.
11. Überprüfen Sie auf der Seite Überprüfen und absenden alle überwachten Komponenten- und Workload-Details.
12. Wählen Sie Absenden aus.

Account-based application monitoring

1. Anwendungsname Geben Sie einen Namen für Ihre kontobasierte Anwendung ein.
2. Automatisierte Überwachung neuer Ressourcen. Standardmäßig verwendet Application Insights empfohlene Einstellungen zum Konfigurieren der Überwachung für Ressourcenkomponenten, die Ihrem Konto nach der Anwendungsintegration hinzugefügt werden. Sie können die Überwachung bei Ressourcen ausschließen, die nach dem Onboarding Ihrer Anwendung hinzugefügt wurden, indem Sie das Kontrollkästchen deaktivieren.
3. CloudWatchEreignisse überwachen. Aktivieren Sie das Kontrollkästchen, um die Überwachung von Application Insights in CloudWatch Events zu integrieren und Einblicke aus Amazon EBS, Amazon EC2, Amazon ECS AWS CodeDeploy, AWS Health APIs And Notifications, Amazon RDS, Amazon S3 und zu erhalten. AWS Step Functions
4. Integrieren Sie mit AWS Systems Manager OpsCenter. Aktivieren Sie das Kontrollkästchen Systems Manager für Abhilfemaßnahmen generieren, um Probleme anzuzeigen und sich benachrichtigen zu lassen, wenn Probleme OpsCenter OpsItems für ausgewählte Anwendungen erkannt werden. Geben Sie das SNS-Thema ARN an, um die Vorgänge nachzuverfolgen, die zur Lösung betrieblicher Arbeitsaufgaben (OpsItems) durchgeführt werden, die sich auf Ihre AWS Ressourcen beziehen.

5. Stichwörter — optional. CloudWatch Application Insights unterstützt sowohl tagbasierte als auch CloudFormation basierte Ressourcengruppen (mit Ausnahme von Auto Scaling Scaling-Gruppen). Weitere Informationen finden Sie unter Arbeiten mit dem Tag Editor.
6. Erkannte Ressourcen. Alle in Ihrem Konto erkannten Ressourcen werden zu dieser Liste hinzugefügt. Wenn Application Insights nicht alle Ressourcen in Ihrem Konto erkennen kann, wird eine Fehlermeldung oben auf der Seite angezeigt. Diese Nachricht enthält einen Link zur Anleitung zum Hinzufügen der erforderlichen Berechtigungen.
7. Wählen Sie Weiter aus.

   Für die Anwendung wird ein ARN im folgenden Format generiert.

   ```
   arn:partition:applicationinsights:region:account-id:application/
   TBD/application-name
   ```

   Beispiel

   ```
   arn:aws:applicationinsights:us-east-1:123456789012:application/TBD/my-
   application
   ```

4. Nachdem Sie die Konfiguration für die Anwendungsüberwachung abgesendet haben, gelangen Sie zur Detailseite für die Anwendung, auf der Sie die Application summary (Anwendungs-Zusammenfassung), die Liste der Monitored components (überwachten Komponenten) und der Unmonitored components (nicht überwachten Komponenten) sehen. Über die Registerkarten neben Components (Komponenten) sehen Sie Configuration history (Konfigurationsverlauf), Log patterns (Protokollmuster) sowie angewandte Tags.

   Um Erkenntnisse für die Anwendung anzuzeigen, wählen Sie View Insights (Erkenntnisse anzeigen) aus.

   Sie können Ihre Auswahl für die Überwachung von CloudWatch Ereignissen und die Integration mit AWS Systems Manager aktualisieren, OpsCenter indem Sie Bearbeiten wählen.

   Unter Components (Komponenten) können Sie im Menü Actions (Aktionen) eine Instance-Gruppe erstellen, ändern oder deren Gruppierung aufheben.

   Sie können die Überwachung für Komponenten verwalten, einschließlich Anwendungsebene, Protokollgruppen, Ereignisprotokolle, Metriken und benutzerdefinierte Alarme, indem Sie

das Aufzählungszeichen neben einer Komponente auswählen und dann Manage monitoring (Überwachung verwalten).

Aktivieren von Application Insights zur Ressourcenüberwachung für Amazon ECS und Amazon EKS

Sie können mit Application Insights containerisierte Anwendungen und Microservices über die Container-Insights-Konsole überwachen. Application Insights unterstützt die Überwachung der folgenden Ressourcen:

- Amazon-ECS-Cluster
- Amazon-ECS-Dienstleistungen
- Amazon-ECS-Aufgaben
- Amazon-EKS-Cluster

Wenn Application Insights aktiviert ist, bietet es empfohlene Metriken und Protokolle, erkennt potenzielle Probleme, generiert CloudWatch Ereignisse und erstellt automatische Dashboards für Ihre containerisierten Anwendungen und Microservices.

Sie können mit Application Insights containerisierte Anwendungen über die Container-Insights- oder die Application-Insights-Konsole überwachen.

Aktivieren Sie Application Insights über die Container-Insights-Konsole

Wählen Sie in der Container Insights-Konsole auf dem Container-Insights-Dashboard Performance monitoring (Leistungsüberwachung) die Option Auto-configure Application Insights (Application Insights automatisch konfigurieren) aus. Wenn Application Insights aktiviert ist, werden Details zu erkannten Problemen angezeigt.

Aktivieren Sie Application Insights über die Application-Insights-Konsole

Wenn ECS-Cluster in der Komponentenliste erscheinen, aktiviert Application Insights automatisch eine zusätzliche Containerüberwachung mit Container Insights.

Für EKS-Cluster können Sie eine zusätzliche Überwachung mit Container Insights aktivieren, um Diagnoseinformationen bereitzustellen, z. B. über fehlgeschlagene Container-Neustarts, wodurch sie Ursachen für Probleme finden und beheben können. Um Container Insights für EKS einzurichten, sind zusätzliche Schritte erforderlich. Weitere Informationen zum Einrichten von Container Insights auf EKS finden Sie unter Einrichten von Container Insights in Amazon EKS und Kubernetes.

Zusätzliche Überwachung für EKS mit Container Insights wird auf Linux-Instances mit EKS unterstützt.

Weitere Informationen zur Unterstützung von Container Insights für ECS- und EKS-Cluster finden Sie unter Container Insights.

Deaktivieren der Überwachung einer Anwendungskomponente

Um die Überwachung für eine Anwendungskomponente zu deaktivieren, wählen Sie auf der Seite mit den Anwendungsdetails die Komponente aus, für die Sie die Überwachung deaktivieren möchten. Wählen Sie Actions (Aktionen) aus und dann Remove from Monitoring (Von der Überwachung ausnehmen).

Löschen einer Anwendung

Um eine Anwendung zu löschen, wählen Sie im CloudWatch Dashboard im linken Navigationsbereich unter Insights die Option Application Insights aus. Wählen Sie die Anwendung aus, die Sie löschen möchten. Wählen Sie unter Actions (Aktionen) die Option Delete application (Anwendung löschen) aus. Dadurch wird die Überwachung gelöscht und alle gespeicherten Überwachungen für Anwendungskomponenten gelöscht. Die Anwendungsressourcen werden nicht gelöscht.

Richten Sie Ihre Anwendung für die Überwachung über die Befehlszeile ein, konfigurieren und verwalten Sie sie

Dieser Abschnitt enthält Schritte zum Einrichten, Konfigurieren und Verwalten Ihrer Anwendung für die Überwachung mithilfe von AWS CLI und AWS Tools for Windows PowerShell.

Befehlszeilenverfahren

- Hinzufügen und Verwalten einer Anwendung
- Verwalten und Aktualisieren der Überwachung
- Konfigurieren der Überwachung für SQL Always On-Verfügbarkeitsgruppen
- Konfigurieren der Überwachung für MySQL RDS
- Konfigurieren der Überwachung für MySQL EC2
- Konfigurieren der Überwachung für PostgreSQL RDS
- Konfigurieren der Überwachung für PostgreSQL EC2
- Konfigurieren der Überwachung für Oracle RDS
- Konfigurieren der Überwachung für Oracle EC2

Hinzufügen und Verwalten einer Anwendung

Sie können Ihre Application Insights-Anwendung mithilfe der Befehlszeile hinzufügen, Informationen darüber abrufen, verwalten und konfigurieren.

Themen

Hinzufügen einer Anwendung

Fügen Sie eine Anwendung hinzu, indem Sie AWS CLI

Verwenden Sie den folgenden AWS CLI Befehl, um eine Anwendung für Ihre Ressourcengruppe namens`my-resource-group`, with OpsCenter enabled to deliver the created OpsItem to the SNS-Thema ARN hinzuzufügen `arn:aws:sns:us-east-1:123456789012:MyTopic`, verwenden Sie den folgenden Befehl.

```
aws application-insights create-application --resource-group-name my-resource-group --
ops-center-enabled --ops-item-sns-topic-arn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Fügen Sie eine Anwendung hinzu mit AWS Tools for Windows PowerShell

Verwenden Sie AWS Tools for Windows PowerShell den folgenden Befehl, um eine Anwendung für Ihre Ressourcengruppe hinzuzufügen, die `my-resource-group` mit OpsCenter enabled aufgerufen wurde, um das erstellte OpsItem an das SNS-Thema ARN `arn:aws:sns:us-east-1:123456789012:MyTopic` zu liefern.

```
New-CWAIApplication -ResourceGroupName my-resource-group -OpsCenterEnabled true -
OpsItemSNSTopicArn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Beschreiben einer Anwendung

Beschreiben Sie eine Anwendung mit dem AWS CLI

Verwenden Sie den folgenden Befehl AWS CLI , um eine Anwendung zu beschreibenmy-resource-group, die für eine Ressourcengruppe namens erstellt wurde.

```
aws application-insights describe-application --resource-group-name my-resource-group
```

Beschreiben Sie eine Anwendung mit AWS Tools for Windows PowerShell

Verwenden Sie den folgenden Befehl AWS Tools for Windows PowerShell , um eine Anwendung zu beschreibenmy-resource-group, die für eine Ressourcengruppe namens erstellt wurde.

```
Get-CWAIApplication -ResourceGroupName my-resource-group
```

Auflisten von Komponenten in einer Anwendung

Listet Komponenten in einer Anwendung auf, indem Sie AWS CLI

Verwenden Sie den folgenden Befehl AWS CLI , um die Komponenten aufzulistenmy-resource-group, die für eine Ressourcengruppe namens erstellt wurden.

```
aws application-insights list-components --resource-group-name my-resource-group
```

Komponenten in einer Anwendung auflisten mit AWS Tools for Windows PowerShell

Verwenden Sie den folgenden Befehl AWS Tools for Windows PowerShell , um die Komponenten aufzulistenmy-resource-group, die für eine aufgerufene Ressourcengruppe erstellt wurden.

```
Get-CWAIComponentList -ResourceGroupName my-resource-group
```

Beschreiben einer Komponente

Beschreiben Sie eine Komponente mit dem AWS CLI

Sie können den folgenden AWS CLI Befehl verwenden, um eine Komponente namens zu beschreibenmy-component, die zu einer Anwendung gehört, die in einer Ressourcengruppe namens erstellt wurdemy-resource-group.

```
aws application-insights describe-component --resource-group-name my-resource-group --
component-name my-component
```

Beschreiben Sie eine Komponente mit AWS Tools for Windows PowerShell

Sie können den folgenden AWS Tools for Windows PowerShell Befehl verwenden, um eine Komponente namens zu beschreibenmy-component, die zu einer Anwendung gehört, die in einer Ressourcengruppe namens erstellt wurdemy-resource-group.

```
Get-CWAIComponent -ComponentName my-component -ResourceGroupName my-resource-group
```

Gruppieren ähnlicher Ressourcen in einer benutzerdefinierten Komponente

Wir empfehlen, ähnliche Ressourcen, wie z. B. .NET-Webserver-Instances, als benutzerdefinierte Komponenten zu gruppieren, um das Onboarding zu erleichtern und die Überwachung und Erkenntnisse zu verbessern. Derzeit unterstützt CloudWatch Application Insights benutzerdefinierte Gruppen für EC2-Instances.

So gruppieren Sie Ressourcen mithilfe der AWS CLI in einer benutzerdefinierten Komponente

Verwenden Sie den folgenden Befehl, AWS CLI um drei Instanzen (arn:aws:ec2:us-east-1:123456789012:instance/i-11111arn:aws:ec2:us-east-1:123456789012:instance/i-22222, und arn:aws:ec2:us-east-1:123456789012:instance/i-33333) zu einer benutzerdefinierten Komponente zu gruppieren, die my-component für eine Anwendung aufgerufen wirdmy-resource-group, die für die aufgerufene Ressourcengruppe erstellt wurde.

```
aws application-insights create-component --resource-group-name my-
resource-group --component-name my-component --resource-list arn:aws:ec2:us-
east-1:123456789012:instance/i-11111 arn:aws:ec2:us-east-1:123456789012:instance/
i-22222 arn:aws:ec2:us-east-1:123456789012:instance/i-33333
```

So gruppieren Sie Ressourcen mithilfe der AWS Tools for Windows PowerShell in einer benutzerdefinierten Komponente

Verwenden Sie AWS Tools for Windows PowerShell den folgenden Befehl, um drei Instanzen (arn:aws:ec2:us-east-1:123456789012:instance/i-11111arn:aws:ec2:us-east-1:123456789012:instance/i-22222, undarn:aws:ec2:us-east-1:123456789012:instance/i-33333) zu einer benutzerdefinierten Komponente

zusammenzufassen`my-component`, die für eine Anwendung, die für die aufgerufene Ressourcengruppe erstellt wurde`my-resource-group`, aufgerufen wird.

```
New-CWAIComponent -ResourceGroupName my-resource-group -ComponentName my-component
 -ResourceList arn:aws:ec2:us-east-1:123456789012:instance/i-11111,arn:aws:ec2:us-
east-1:123456789012:instance/i-22222,arn:aws:ec2:us-east-1:123456789012:instance/
i-33333
```

Aufheben der Gruppierung einer benutzerdefinierten Komponente

Um die Gruppierung einer benutzerdefinierten Komponente aufzuheben, verwenden Sie den AWS CLI

Verwenden Sie den folgenden AWS CLI Befehl, um die Gruppierung einer benutzerdefinierten Komponente aufzuheben, die `my-component` in einer Anwendung benannt ist`my-resource-group`, die für die Ressourcengruppe erstellt wurde.

```
aws application-insights delete-component --resource-group-name my-resource-group --
component-name my-new-component
```

Um die Gruppierung einer benutzerdefinierten Komponente aufzuheben, verwenden Sie AWS Tools for Windows PowerShell

Verwenden Sie den folgenden AWS Tools for Windows PowerShell Befehl, um die Gruppierung einer benutzerdefinierten Komponente aufzuheben, die `my-component` in einer Anwendung benannt ist`my-resource-group`, die für die Ressourcengruppe erstellt wurde.

```
Remove-CWAIComponent -ComponentName my-component -ResourceGroupName my-resource-group
```

Aktualisieren einer Anwendung

Aktualisieren Sie eine Anwendung mit dem AWS CLI

Sie können die verwenden AWS CLI , um eine Anwendung zu aktualisieren, AWS Systems Manager OpsCenter OpsItems für Probleme zu generieren, die mit der Anwendung erkannt wurden, und um das erstellte Thema mit dem SNS-Thema OpsItems zu verknüpfen `arn:aws:sns:us-east-1:123456789012:MyTopic`, indem Sie den folgenden Befehl verwenden.

```
aws application-insights update-application --resource-group-name my-resource-group --
ops-center-enabled --ops-item-sns-topic-arn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Aktualisieren Sie eine Anwendung mithilfe von AWS Tools für Windows PowerShell

Mithilfe des AWS Tools for Windows PowerShell folgenden Befehls können Sie eine Anwendung aktualisieren, um AWS SSM OpsCenter OpsItems für Probleme zu generieren, die mit der Anwendung erkannt wurden, und OpsItems die erstellten Dateien dem SNS-Thema `arn:aws:sns:us-east-1:123456789012:MyTopic` zuordnen.

```
Update-CWAIApplication -ResourceGroupName my-resource-group -OpsCenterEnabled true -
OpsItemSNSTopicArn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Aktualisieren einer benutzerdefinierten Komponente

Aktualisieren Sie eine benutzerdefinierte Komponente mit dem AWS CLI

Mithilfe des AWS CLI folgenden Befehls können Sie eine benutzerdefinierte Komponente `my-component` mit einem neuen Komponentennamen und einer aktualisierten Gruppe von Instanzen aktualisieren. `my-new-component`

```
aws application-insights update-component --resource-group-name my-resource-
group --component-name my-component --new-component-name my-new-component --
resource-list arn:aws:ec2:us-east-1:123456789012:instance/i-44444 arn:aws:ec2:us-
east-1:123456789012:instance/i-55555
```

Aktualisieren Sie eine benutzerdefinierte Komponente mithilfe von AWS Tools für Windows PowerShell

Mithilfe des AWS Tools for Windows PowerShell folgenden Befehls können Sie eine benutzerdefinierte Komponente `my-component` mit einem neuen Komponentennamen und einer aktualisierten Instanzgruppe aktualisieren. `my-new-component`

```
Update-CWAIComponent -ComponentName my-component -NewComponentName my-new-
component -ResourceGroupName my-resource-group -ResourceList arn:aws:ec2:us-
east-1:123456789012:instance/i-44444,arn:aws:ec2:us-east-1:123456789012:instance/
i-55555
```

Verwalten und Aktualisieren der Überwachung

Sie können die Überwachung Ihrer Application Insights-Anwendung mithilfe der Befehlszeile verwalten und aktualisieren.

Themen

Auflisten von Problemen mit Ihrer Anwendung

Führen Sie Probleme mit Ihrer Anwendung auf, indem Sie AWS CLI

Verwenden Sie den folgenden AWS CLI Befehl, um Probleme mit Ihrer Anwendung aufzulisten, die zwischen 1.000 und 10.000 Millisekunden seit der Unix-Epoche für eine Anwendung entdeckt wurden`my-resource-group`, die in einer Ressourcengruppe namens erstellt wurde.

```
aws application-insights list-problems --resource-group-name my-resource-group --start-
time 1000 --end-time 10000
```

Führen Sie mithilfe von Tools für Windows Probleme mit Ihrer Anwendung auf AWS PowerShell

Verwenden Sie den folgenden AWS Tools for Windows PowerShell Befehl, um Probleme mit Ihrer Anwendung aufzulisten, die zwischen 1.000 und 10.000 Millisekunden seit der Unix-Epoche für eine Anwendung erkannt wurden`my-resource-group`, die auf einer aufgerufenen Ressourcengruppe erstellt wurde.

```
$startDate = "8/6/2019 3:33:00"
$endDate = "8/6/2019 3:34:00"
Get-CWAIProblemList -ResourceGroupName my-resource-group -StartTime $startDate -
EndTime $endDate
```

Beschreiben eines Anwendungsproblems

Beschreiben Sie ein Anwendungsproblem mit dem AWS CLI

Verwenden Sie den folgenden Befehlp-1234567890, AWS CLI um ein Problem mit der Problem-ID zu beschreiben.

```
aws application-insights describe-problem –problem-id p-1234567890
```

Beschreiben Sie ein Anwendungsproblem mithilfe von AWS Tools für Windows PowerShell

Verwenden Sie den folgenden Befehlp-1234567890, AWS Tools for Windows PowerShell um ein Problem mit der ID zu beschreiben.

```
Get-CWAIProblem -ProblemId p-1234567890
```

Beschreiben der mit einem Problem verbundenen Anomalien oder Fehler

Beschreiben Sie die mit einem Problem verbundenen Anomalien oder Fehler mithilfe der AWS CLI

Verwenden Sie den folgenden AWS CLI Befehl, um die Anomalien oder Fehler im Zusammenhang mit einem Problem mit der Problem-ID p-1234567890 zu beschreiben.

```
aws application-insights describe-problem-observations --problem-id -1234567890
```

Beschreiben der mit einem Problem verbundenen Anomalien oder Fehler mithilfe der AWS Tools for Windows PowerShell

Verwenden Sie den folgenden AWS Tools for Windows PowerShell Befehl, um die Anomalien oder Fehler im Zusammenhang mit einem Problem mit der Problem-ID p-1234567890 zu beschreiben.

```
Get-CWAIProblemObservation -ProblemId p-1234567890
```

Beschreiben einer Anomalie oder eines Fehlers mit der Anwendung

Beschreiben einer Anomalie oder eines Fehlers mit der Anwendung mithilfe der AWS CLI

Verwenden Sie den folgenden AWS CLI Befehl, um eine Anomalie oder einen Fehler in der Anwendung mit der Beobachtungs-ID o-1234567890 zu beschreiben.

```
aws application-insights describe-observation –observation-id o-1234567890
```

Beschreiben Sie mithilfe von AWS Tools für Windows eine Anomalie oder einen Fehler in der Anwendung PowerShell

Verwenden Sie den folgenden Befehl AWS Tools for Windows PowerShell , um eine Anomalie oder einen Fehler in der Anwendung mit der Beobachtungs-ID `o-1234567890` zu beschreiben.

```
Get-CWAIObservation -ObservationId o-1234567890
```

Beschreiben der Überwachungskonfigurationen einer Komponente

Beschreiben der Überwachungskonfigurationen einer Komponente mithilfe der AWS CLI

Verwenden Sie den folgenden Befehl AWS CLI , um die Überwachungskonfiguration einer Komponente zu beschreiben, die `my-component` in einer Anwendung aufgerufen wurde`my-resource-group`, die für die Ressourcengruppe erstellt wurde.

```
aws application-insights describe-component-configuration –resource-group-name my-resource-group –component-name my-component
```

Beschreiben Sie die Überwachungskonfigurationen einer Komponente mithilfe von AWS Tools für Windows PowerShell

Verwenden Sie den folgenden Befehl`my-component`, AWS Tools for Windows PowerShell um die Überwachungskonfiguration einer aufgerufenen Komponente in einer Anwendung zu beschreiben`my-resource-group`, die auf der Ressourcengruppe erstellt wurde.

```
Get-CWAIComponentConfiguration -ComponentName my-component -ResourceGroupName my-resource-group
```

Weitere Informationen zur Komponentenkonfiguration und JSON-Beispieldateien finden Sie unter Arbeiten mit Komponentenkonfigurationen.

Beschreiben der empfohlenen Überwachungskonfiguration einer Komponente

Beschreiben Sie die empfohlene Überwachungskonfiguration einer Komponente mithilfe des AWS CLI

Wenn die Komponente Teil einer .NET Worker-Anwendung ist, können Sie die verwenden, AWS CLI um die empfohlene Überwachungskonfiguration einer Komponente zu beschreiben,

die `my-component` in einer Anwendung aufgerufen wird`my-resource-group`, die für die Ressourcengruppe erstellt wurde. Verwenden Sie dazu den folgenden Befehl.

```
aws application-insights describe-component-configuration-recommendation --resource-
group-name my-resource-group --component-name my-component --tier DOT_NET_WORKER
```

Beschreiben Sie die empfohlene Überwachungskonfiguration einer Komponente mit AWS Tools for Windows PowerShell

Wenn die Komponente Teil einer .NET Worker-Anwendung ist, können Sie die verwenden, AWS Tools for Windows PowerShell um die empfohlene Überwachungskonfiguration einer Komponente zu beschreiben, die `my-component` in einer Anwendung aufgerufen wird`my-resource-group`, die für die Ressourcengruppe erstellt wurde. Verwenden Sie dazu den folgenden Befehl.

```
Get-CWAIComponentConfigurationRecommendation -ComponentName my-component -
ResourceGroupName my-resource-group -Tier DOT_NET_WORKER
```

Weitere Informationen zur Komponentenkonfiguration und JSON-Beispieldateien finden Sie unter Arbeiten mit Komponentenkonfigurationen.

Aktualisieren der Überwachungskonfigurationen für eine Komponente

Aktualisieren der Überwachungskonfigurationen für eine Komponente mithilfe der AWS CLI

Verwenden Sie den folgenden Befehl AWS CLI , um die Komponente zu aktualisieren, die `my-component` in einer Anwendung aufgerufen wurde`my-resource-group`, die für die aufgerufene Ressourcengruppe erstellt wurde. Der Befehl umfasst die folgenden Aktionen:

1. Aktivieren Sie die Überwachung für die Komponente.
2. Legen Sie die Ebene der Komponente auf `.NET Worker` fest.
3. Aktualisieren Sie die JSON-Konfiguration der Komponente, um aus der lokalen Datei `configuration.txt` zu lesen.

```
aws application-insights update-component-configuration --resource-group-name my-
resource-group --component-name my-component --tier DOT_NET_WORKER --monitor --
component-configuration "file://configuration.txt"
```

Aktualisieren der Überwachungskonfigurationen für eine Komponente mithilfe der AWS Tools for Windows PowerShell

Verwenden Sie den folgenden Befehl AWS Tools for Windows PowerShell , um die Komponente zu aktualisieren, die `my-component` in einer Anwendung aufgerufen wurde`my-resource-group`, die für die aufgerufene Ressourcengruppe erstellt wurde. Der Befehl umfasst die folgenden Aktionen:

1. Aktivieren Sie die Überwachung für die Komponente.
2. Legen Sie die Ebene der Komponente auf `.NET Worker` fest.
3. Aktualisieren Sie die JSON-Konfiguration der Komponente, um aus der lokalen Datei `configuration.txt` zu lesen.

```
[string]$config = Get-Content -Path configuration.txt
Update-CWAIComponentConfiguration -ComponentName my-component -ResourceGroupName my-
resource-group -Tier DOT_NET_WORKER -Monitor 1 -ComponentConfiguration $config
```

Weitere Informationen zur Komponentenkonfiguration und JSON-Beispieldateien finden Sie unter Arbeiten mit Komponentenkonfigurationen.

Entfernen einer angegebenen Ressourcengruppe aus der Application-Insights-Überwachung

Entfernen Sie eine angegebene Ressourcengruppe aus der Application Insights-Überwachung mithilfe des AWS CLI

Verwenden Sie den folgenden Befehl AWS CLI , um eine Anwendung, die für die aufgerufene Ressourcengruppe erstellt wurde, `my-resource-group` aus der Überwachung zu entfernen.

```
aws application-insights delete-application --resource-group-name my-resource-group
```

Entfernen Sie eine angegebene Ressourcengruppe aus der Application Insights-Überwachung mithilfe des AWS Tools for Windows PowerShell

Verwenden Sie den folgenden Befehl AWS Tools for Windows PowerShell , um eine Anwendung, die für die aufgerufene Ressourcengruppe erstellt wurde, `my-resource-group` aus der Überwachung zu entfernen.

```
Remove-CWAIApplication -ResourceGroupName my-resource-group
```

Konfigurieren der Überwachung für SQL Always On-Verfügbarkeitsgruppen

1. Erstellen Sie eine Anwendung für die Ressourcengruppe mit den SQL HA-EC2-Instances.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Definieren Sie die EC2-Instances, die den SQL-HA-Cluster darstellen, indem Sie eine neue Anwendungskomponente erstellen.

```
aws application-insights create-component --resource-group-name
 "<RESOURCE_GROUP_NAME>" --component-name SQL_HA_CLUSTER --resource-list
  "arn:aws:ec2:<REGION>:<ACCOUNT_ID>:instance/<CLUSTER_INSTANCE_1_ID>"
 "arn:aws:ec2:<REGION>:<ACCOUNT_ID>:instance/<CLUSTER_INSTANCE_2_ID>
```

3. Konfigurieren Sie die SQL-HA-Komponente.

```
aws application-insights  update-component-configuration --resource-group-name
 "<RESOURCE_GROUP_NAME>" --region <REGION> --component-name "SQL_HA_CLUSTER" --
monitor --tier SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP --monitor  --component-
configuration '{
  "subComponents" : [ {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [ {
      "alarmMetricName" : "CPUUtilization",
      "monitor" : true
    }, {
      "alarmMetricName" : "StatusCheckFailed",
      "monitor" : true
    }, {
      "alarmMetricName" : "Processor % Processor Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory % Committed Bytes In Use",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory Available Mbytes",
      "monitor" : true
    }, {
      "alarmMetricName" : "Paging File % Usage",
      "monitor" : true
    }, {
      "alarmMetricName" : "System Processor Queue Length",
      "monitor" : true
    }, {
      "alarmMetricName" : "Network Interface Bytes Total/sec",
```

```
      "monitor" : true
    }, {
      "alarmMetricName" : "PhysicalDisk % Disk Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Buffer cache hit ratio",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Page life expectancy",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:General Statistics Processes blocked",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:General Statistics User Connections",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Locks Number of Deadlocks/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:SQL Statistics Batch Requests/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica File Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log remaining for undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Send Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Mirrored Write Transaction/
sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Recovery Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redo Bytes Remaining",
      "monitor" : true
```

```
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redone Bytes/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Total Log requiring undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Transaction Delay",
      "monitor" : true
    } ],
    "windowsEvents" : [ {
      "logGroupName" : "WINDOWS_EVENTS-Application-<RESOURCE_GROUP_NAME>",
      "eventName" : "Application",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL", "INFORMATION" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-System-<RESOURCE_GROUP_NAME>",
      "eventName" : "System",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-Security-<RESOURCE_GROUP_NAME>",
      "eventName" : "Security",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    } ],
    "logs" : [ {
      "logGroupName" : "SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP-
<RESOURCE_GROUP_NAME>",
      "logPath" : "C:\\Program Files\\Microsoft SQL Server\\MSSQL**.MSSQLSERVER\
\MSSQL\\Log\\ERRORLOG",
      "logType" : "SQL_SERVER",
      "monitor" : true,
      "encoding" : "utf-8"
    } ]
  }, {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [ {
      "alarmMetricName" : "VolumeReadBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteBytes",
      "monitor" : true
    }, {
```

```
    "alarmMetricName" : "VolumeReadOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeQueueLength",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeThroughputPercentage",
      "monitor" : true
    }, {
    "alarmMetricName" : "BurstBalance",
      "monitor" : true
    } ]
  } ]
}'
```

> **Note**
>
> Application Insights muss Anwendungsereignisprotokolle (Informationsebene) aufnehmen, um Clusteraktivitäten wie Failover zu erkennen.

Konfigurieren der Überwachung für MySQL RDS

1. Erstellen Sie eine Anwendung für die Ressourcengruppe mit der RDS MySQL-Datenbank-Instance.

   ```
   aws application-insights create-application --region <REGION> --resource-group-name
     <RESOURCE_GROUP_NAME>
   ```

2. Das Fehlerprotokoll ist standardmäßig aktiviert. Das Slow Query-Protokoll kann mithilfe von Datenparametergruppen aktiviert werden. Weitere Informationen finden Sie unter Zugriff auf die MySQL Slow Query- und allgemeinen Protokolle.
   - `set slow_query_log = 1`
   - `set log_output = FILE`
3. Exportieren Sie die zu überwachenden CloudWatch Protokolle in Protokolle. Weitere Informationen finden Sie unter MySQL-Logs in CloudWatch Logs veröffentlichen.

4. Konfigurieren Sie die MySQL RDS-Komponente.

```
aws application-insights  update-component-configuration --resource-group-name
 "<RESOURCE_GROUP_NAME>" --region <REGION> --component-name "<DB_COMPONENT_NAME>"
 --monitor --tier DEFAULT --monitor  --component-configuration "{\"alarmMetrics\":
[{\"alarmMetricName\":\"CPUUtilization\",\"monitor\":true}],\"logs\":[{\"logType\":
\"MYSQL\",\"monitor\":true},{\"logType\": \"MYSQL_SLOW_QUERY\",\"monitor\":false}]}"
```

Konfigurieren der Überwachung für MySQL EC2

1. Erstellen Sie eine Anwendung für die Ressourcengruppe mit den SQL HA-EC2-Instances.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Das Fehlerprotokoll ist standardmäßig aktiviert. Das Slow Query-Protokoll kann mithilfe von Datenparametergruppen aktiviert werden. Weitere Informationen finden Sie unter Zugriff auf die MySQL Slow Query- und allgemeinen Protokolle.
   - `set slow_query_log = 1`
   - `set log_output = FILE`
3. Konfigurieren Sie die MySQL EC2-Komponente.

```
aws application-insights  update-component-configuration --resource-group-name
 "<RESOURCE_GROUP_NAME>" --region <REGION> --component-name "<DB_COMPONENT_NAME>"
 --monitor --tier MYSQL --monitor  --component-configuration "{\"alarmMetrics\":
[{\"alarmMetricName\":\"CPUUtilization\",\"monitor\":true}],\"logs\":[{\"logGroupName
\":\"<UNIQUE_LOG_GROUP_NAME>\",\"logPath\":\"C:\\\\ProgramData\\\\MySQL\\\\MySQL
 Server **\\\\Data\\\\<FILE_NAME>.err\",\"logType\":\"MYSQL\",\"monitor\":true,
\"encoding\":\"utf-8\"}]}"
```

Konfigurieren der Überwachung für PostgreSQL RDS

1. Erstellen Sie eine Anwendung für die Ressourcengruppe mit der PostgreSQL-RDS-Datenbank-Instance.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Das Veröffentlichen von PostgreSQL-Protokollen in CloudWatch ist standardmäßig nicht aktiviert. Um die Überwachung zu aktivieren, öffnen Sie die RDS-Konsole und wählen die zu überwachende Datenbank aus. Wählen Sie in der oberen rechten Ecke Modify (Ändern) aus und aktivieren Sie das Kontrollkästchen für das PostgreSQL-Protokoll. Wählen Sie Continue (Weiter) aus, um diese Einstellung zu speichern.
3. Ihre PostgreSQL-Logs werden nach exportiert. CloudWatch
4. Konfigurieren Sie PostgreSQL-RDS-Komponente.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier DEFAULT --component-configuration
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\": \"CPUUtilization\",
         \"monitor\": true
      }
   ],
   \"logs\":[
      {
         \"logType\": \"POSTGRESQL\",
         \"monitor\": true
      }
   ]
}"
```

Konfigurieren der Überwachung für PostgreSQL EC2

1. Erstellen Sie eine Anwendung für die Ressourcengruppe mit den PostgreSQL-EC2-Instances.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Konfigurieren Sie die Komponente PostgreSQL EC2.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier POSTGRESQL --component-configuration
"{
   \"alarmMetrics\":[
```

```
      {
         \"alarmMetricName\":\"CPUUtilization\",
         \"monitor\":true
      }
   ],
   \"logs\":[
      {
         \"logGroupName\":\"<UNIQUE_LOG_GROUP_NAME>\",
         \"logPath\":\"/var/lib/pgsql/data/log/\",
         \"logType\":\"POSTGRESQL\",
         \"monitor\":true,
         \"encoding\":\"utf-8\"
      }
   ]
}"
```

Konfigurieren der Überwachung für Oracle RDS

1. Erstellen Sie eine Anwendung für die Ressourcengruppe mit der Oracle-RDS-Datenbank-Instance.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Das Veröffentlichen von Oracle-Protokollen auf CloudWatch ist standardmäßig nicht aktiviert. Um die Überwachung zu aktivieren, öffnen Sie die RDS-Konsole und wählen die zu überwachende Datenbank aus. Wählen Sie in der oberen rechten Ecke Modify (Ändern) aus und aktivieren Sie die Kontrollkästchen mit den Bezeichnungen Alert (Warnungs)-Protokoll und Listener-Protokoll. Wählen Sie Continue (Weiter) aus, um diese Einstellung zu speichern.
3. Ihre Oracle-Logs werden nach exportiert CloudWatch.
4. Konfigurieren Sie die Oracle-RDS-Komponente.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier DEFAULT --component-configuration
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\": \"CPUUtilization\",
         \"monitor\": true
      }
```

```
   ],
   \"logs\":[
      {
         \"logType\": \"ORACLE_ALERT\",
         \"monitor\": true
      },
      {
         \"logType\": \"ORACLE_LISTENER\",
         \"monitor\": true
      }
   ]
}"
```

Konfigurieren der Überwachung für Oracle EC2

1. Erstellen Sie eine Anwendung für die Ressourcengruppe mit den Oracle-EC2-Instance.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Konfigurieren Sie die Oracle-EC2-Komponente.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier ORACLE --component-configuration
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\":\"CPUUtilization\",
         \"monitor\":true
      }
   ],
   \"logs\":[
      {
         \"logGroupName\":\"<UNIQUE_LOG_GROUP_NAME>\",
         \"logPath\":\"/opt/oracle/diag/rdbms/*/*/trace\",
         \"logType\":\"ORACLE_ALERT\",
         \"monitor\":true,
      },
      {
         \"logGroupName\":\"<UNIQUE_LOG_GROUP_NAME>\",
         \"logPath\":\"/opt/oracle/diag/tnslsnr/$HOSTNAME/listener/trace/\",
```

```
        \"logType\":\"ORACLE_ALERT\",
        \"monitor\":true,
      }
   ]
}"
```

Application Insights — CloudWatch Ereignisse und Benachrichtigungen bei erkannten Problemen

Für jede Anwendung, die zu CloudWatch Application Insights hinzugefügt wird, wird nach bestem Wissen und Gewissen ein CloudWatch Ereignis für die folgenden Ereignisse veröffentlicht:

- Problemerstellung Wird ausgelöst, wenn CloudWatch Application Insights ein neues Problem erkennt.
  - Detailtyp: "Application Insights Problem erkannt"
  - Detail:
    - `problemId`: Die ID des erkannten Problems.
    - `region`: Die AWS Region, in der das Problem verursacht wurde.
    - `resourceGroupName`: Die Ressourcengruppe der registrierten Anwendung, für die das Problem erkannt wurde.
    - `status`: Der Status des Problems. Die möglichen Status und Definitionen lauten wie folgt:
      - `In progress`: Ein neues Problem wurde identifiziert. Das Problem wird immer noch beobachtet.
      - `Recovering`: Das Problem stabilisiert sich. Sie können das Problem manuell lösen, wenn es sich in diesem Zustand befindet.
      - `Resolved`: Das Problem wurde gelöst. Zu diesem Problem gibt es keine neuen Beobachtungen.
      - `Recurring`: Das Problem wurde innerhalb der letzten 24 Stunden behoben. Es wurde aufgrund zusätzlicher Beobachtungen wiedereröffnet.
    - `severity`: Der Schweregrad des Problems.
    - `problemUrl`: Die Konsolen-URL für das Problem.

- Problemaktualsierung Entfällt, wenn das Problem mit einer neuen Beobachtung aktualisiert wird oder wenn eine bestehende Beobachtung aktualisiert wird und das Problem anschließend aktualisiert wird; Updates beinhalten eine Behebung oder Schließung des Problems.

- Detailtyp: „Application Insights-Problem aktualisiert“
- Detail:
  - `problemId`: Die ID des erstellten Problems.
  - `region`: Die AWS Region, in der das Problem verursacht wurde.
  - `resourceGroupName`: Die Ressourcengruppe der registrierten Anwendung, für die das Problem erkannt wurde.
  - `status`: Der Status des Problems.
  - `severity`: Der Schweregrad des Problems.
  - `problemUrl`: Die Konsolen-URL für das Problem.

So erhalten Sie Benachrichtigungen über Problemereignisse, die von einer Anwendung erzeugt werden

Wählen Sie in der CloudWatch Konsole im linken Navigationsbereich unter Ereignisse die Option Regeln aus. Wählen Sie auf der Seite Rules (Regeln) die Option Create rule (Regel erstellen) aus. Wählen Sie Amazon CloudWatch Application Insights aus der Dropdownliste Service Name und wählen Sie den Ereignistyp aus. Wählen Sie dann Add target (Ziel hinzufügen) und das Ziel und die Parameter aus, z. B. ein SNS topic (SNS-Thema) oder Lambda function (Lambda-Funktion).

Aktionen bis AWS Systems Manager. CloudWatch Application Insights bietet eine integrierte Integration mit Systems Manager OpsCenter. Wenn Sie sich dafür entscheiden, diese Integration für Ihre Anwendung zu verwenden, OpsItem wird auf der OpsCenter Konsole für jedes Problem, das mit der Anwendung erkannt wird, eine erstellt. In der OpsCenter Konsole können Sie zusammengefasste Informationen zu dem von CloudWatch Application Insights erkannten Problem anzeigen und ein Systems Manager Automation-Runbook auswählen, um Abhilfemaßnahmen zu ergreifen oder Windows-Prozesse, die Ressourcenprobleme in Ihrer Anwendung verursachen, genauer zu identifizieren.

Kontoübergreifende Beobachtbarkeit von Application Insights

Mit der kontenübergreifenden Beobachtbarkeit von CloudWatch Application Insights können Sie Ihre Anwendungen überwachen und Fehler beheben, die sich über mehrere AWS Konten innerhalb einer einzigen Region erstrecken.

Sie können Amazon CloudWatch Observability Access Manager verwenden, um eines oder mehrere Ihrer AWS Konten als Überwachungskonto einzurichten. Sie geben dem Überwachungskonto

die Möglichkeit, die Daten in Ihrem Quellkonto einzusehen, indem Sie eine Sink in Ihrem Überwachungskonto erstellen. Sie verwenden die Sink, um eine Verbindung von Ihrem Quellkonto zu Ihrem Überwachungskonto herzustellen. Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.

Erforderliche -Ressourcen

Damit die kontenübergreifende Observability von CloudWatch Application Insights ordnungsgemäß funktioniert, stellen Sie sicher, dass die folgenden Telemetriearten über den CloudWatch Observability Access Manager gemeinsam genutzt werden.

- Anwendungen in Application Insights CloudWatch
- Metriken bei Amazon CloudWatch
- Gruppen in Amazon CloudWatch Logs protokollieren
- Ablaufverfolgungen in AWS X-Ray

Arbeiten mit Komponentenkonfigurationen

Eine Komponentenkonfiguration ist eine Textdatei im JSON-Format, die die Konfigurationseinstellungen der Komponente beschreibt. Dieser Abschnitt enthält ein Beispielvorlagenfragment, Beschreibungen von Abschnitten zur Komponentenkonfiguration und Beispielkonfigurationen.

Themen

- Fragment einer Komponentenkonfiguration
- Abschnitte einer Komponentenkonfiguration
- Beispiele für die Komponentenkonfiguration

Fragment einer Komponentenkonfiguration

Das folgende Beispiel zeigt ein Vorlagenfragment im JSON-Format.

```
{
  "alarmMetrics" : [
    list of alarm metrics
  ],
  "logs" : [
    list of logs
```

```
  ],
  "processes" : [
   list of processes
  ],
  "windowsEvents" : [
    list of windows events channels configurations
  ],
  "alarms" : [
    list of CloudWatch alarms
  ],
  "jmxPrometheusExporter": {
    JMX Prometheus Exporter configuration
  },
  "hanaPrometheusExporter": {
      SAP HANA Prometheus Exporter configuration
  },
  "haClusterPrometheusExporter": {
      HA Cluster Prometheus Exporter configuration
  },
  "netWeaverPrometheusExporter": {
      SAP NetWeaver Prometheus Exporter configuration
  },
  "subComponents" : [
    {
      "subComponentType" : "AWS::EC2::Instance" ...
      component nested instances configuration
    },
    {
      "subComponentType" : "AWS::EC2::Volume" ...
      component nested volumes configuration
    }
  ]
}
```

Abschnitte einer Komponentenkonfiguration

Die Komponentenkonfiguration umfasst mehrere Hauptabschnitte. Abschnitte in einer Komponentenkonfiguration können in beliebiger Reihenfolge aufgelistet sein.

- alarmMetrics (optional)

  Eine Liste der Metriken, die für die Komponente überwacht werden sollen. Alle Komponentententypen können über einen alarmMetrics-Abschnitt verfügen.

- logs (optional)

  Eine Liste der Protokolle, die für die Komponente überwacht werden sollen. Nur EC2-Instances können über einen logs-Abschnitt verfügen.

- bearbeitet (optional)

  Eine Liste der Prozesse, die für die Komponente überwacht werden sollen. Nur EC2-Instances können über einen Prozessabschnitt verfügen.

- subComponents (optional)

  Verschachtelte Instance- und Volume-SubComponent-Konfiguration für die Komponente. Die folgenden Komponententypen können verschachtelte Instances und einen subComponents-Abschnitt haben: ELB, ASG, benutzerdefinierte gruppierte EC2-Instances und EC2-Instances.

- Alarme (optional)

  Eine Liste der Alarme, die für die Komponente überwacht werden sollen. Alle Komponententypen können einen Alarmbereich haben.

- windowsEvents (optional)

  Eine Liste der Windows-Ereignisse, die für die Komponente überwacht werden sollen. Nur Windows auf EC2-Instances haben einen `windowsEvents`-Abschnitt.

- JMX PrometheusExporter (optional)

  JMXPrometheus-Exporter-Konfiguration.

- hanaPrometheusExporter (fakultativ)

  Konfiguration des SAP HANA-Prometheus-Exporters.

- haClusterPrometheusExporteur (fakultativ)

  Konfiguration des Cluster Prometheus Exporter (HA).

- netWeaverPrometheusExporteur (fakultativ)

  Konfiguration des SAP NetWeaver Prometheus Exporters.

- sapAsePrometheusExporteur (optional)

  Konfiguration des SAP ASE Prometheus Exporters.

Das folgende Beispiel zeigt das die Syntax für das subComponents-Abschnittsfragment im JSON-Format.

```
[
  {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [
      list of alarm metrics
    ],
    "logs" : [
      list of logs
    ],
    "processes": [
      list of processes
    ],
    "windowsEvents" : [
      list of windows events channels configurations
    ]
  },
  {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [
      list of alarm metrics
    ]
  }
]
```

Eigenschaften des Abschnitts Komponentenkonfiguration

In diesem Abschnitt werden die Eigenschaften der einzelnen Komponentenkonfigurationsabschnitte beschrieben.

Sections

- Metrik
- Protokoll
- Prozess
- JMX Prometheus Exporter
- HANA Prometheus Exporter
- HA-Cluster-Prometheus-Exporteur

- NetWeaver Prometheus-Exporteur
- SAP ASE Prometheus Exporter
- Windows-Ereignisse
- Alarm

Metrik

Definiert eine Metrik, die für die Komponente überwacht werden soll.

JSON

```
{
  "alarmMetricName" : "monitoredMetricName",
  "monitor" : true/false
}
```

Eigenschaften

- alarmMetricName (erforderlich)

  Der Name der Metrik, die für die Komponente überwacht werden soll. Informationen zu Metriken, die von Application Insights unterstützt werden, finden Sie unter Von Amazon CloudWatch Application Insights unterstützte Protokolle und Metriken.

- monitor (optional)

  Boolescher Wert, der angibt, ob die Metrik überwacht werden soll. Der Standardwert ist `true`.

Protokoll

Definiert ein Protokoll, das für die Komponente überwacht werden soll.

JSON

```
{
  "logGroupName" : "logGroupName",
  "logPath" : "logPath",
  "logType" : "logType",
  "encoding" : "encodingType",
```

```
  "monitor" : true/false
}
```

Eigenschaften

- logGroupName (erforderlich)

  Der Name der CloudWatch Protokollgruppe, die dem überwachten Protokoll zugeordnet werden soll. Informationen zu den Einschränkungen für Protokollgruppennamen finden Sie unter CreateLogGroup.
- LogPath (erforderlich für EC2-Instanzkomponenten; nicht erforderlich für Komponenten, die den CloudWatch Agenten nicht verwenden, z. B.) AWS Lambda

  Der Pfad der zu überwachenden Protokolle. Der Protokollpfad muss ein absoluter Windows-Systemdateipfad sein. Weitere Informationen finden Sie unter CloudWatch Agent-Konfigurationsdatei: Abschnitt Protokolle.
- logType (erforderlich)

  Der Protokolltyp bestimmt die Protokollmuster, anhand derer Application Insights das Protokoll analysiert. Der Protokolltyp wird aus den folgenden Optionen ausgewählt:
  - SQL_SERVER
  - MYSQL
  - MYSQL_SLOW_QUERY
  - POSTGRESQL
  - ORACLE_ALERT
  - ORACLE_LISTENER
  - IIS
  - APPLICATION
  - WINDOWS_EVENTS
  - WINDOWS_EVENTS_ACTIVE_DIRECTORY
  - WINDOWS_EVENTS_DNS
  - WINDOWS_EVENTS_IIS
  - WINDOWS_EVENTS_SHAREPOINT
  - SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP
  - SQL_SERVER_FAILOVER_CLUSTER_INSTANCE

  - DEFAULT
  - CUSTOM
  - STEP_FUNCTION
  - API_GATEWAY_ACCESS
  - API_GATEWAY_EXECUTION
  - SAP_HANA_LOGS
  - SAP_HANA_TRACE
  - SAP_HANA_HIGH_AVAILABILITY
  - SAP_NETWEAVER_DEV_TRACE_LOGS
  - PACEMAKER_HIGH_AVAILABILITY
- encoding (optional)

  Der Typ der Codierung der zu überwachenden Protokolle. Die angegebene Kodierung sollte in der Liste der vom CloudWatch Agenten unterstützten Kodierungen enthalten sein. Falls nicht angegeben, verwendet CloudWatch Application Insights die Standardkodierung vom Typ utf-8, mit Ausnahme von:

  - SQL_SERVER: utf-16-Verschlüsselung
  - IIS: ascii-Verschlüsselung
- monitor (optional)

  Boolescher Wert, der angibt, ob die Protokolle überwacht werden sollen. Der Standardwert ist `true`.

Prozess

Definiert einen Prozess, der für die Komponente überwacht werden soll.

JSON

```
{
  "processName" : "monitoredProcessName",
  "alarmMetrics" : [
      list of alarm metrics
  ]
}
```

Eigenschaften

- processName (erforderlich)

  Der Name des Prozesses, der für die Komponente überwacht werden soll. Der Prozessname darf keinen Prozessstamm enthalten, wie `sqlservr` oder `sqlservr.exe`.

- alarmMetrics (erforderlich)

  Eine Liste von Metriken, um diesen Prozess zu überwachen. Informationen zur Anzeige von Prozessmetriken, die von CloudWatch Application Insights unterstützt werden, finden Sie unter. Amazon Elastic Compute Cloud (EC2)

JMX Prometheus Exporter

Definiert die JMX-Prometheus-Exporter-Einstellungen.

JSON

```
"JMXPrometheusExporter": {
  "jmxURL" : "JMX URL",
  "hostPort" : "The host and port",
  "prometheusPort" : "Target port to emit Prometheus metrics"
}
```

Eigenschaften

- jmxURL (optional)

  Eine vollständige JMX-URL für die Verbindung.

- hostPort (optional)

  Der Host und Port für die Verbindung über Remote-JMX. Es kann nur einer von `jmxURL` und `hostPort` angegeben werden.

- prometheusPort (optional)

  Der Ziel-Port, an den Prometheus-Metriken gesendet werden sollen. Wenn nicht angegeben, wird der Standard-Port 9404 verwendet.

HANA Prometheus Exporter

Definiert die HANA-Prometheus-Exporter-Einstellungen.

JSON

```
"hanaPrometheusExporter": {
    "hanaSid": "SAP HANA  SID",
    "hanaPort": "HANA database port",
    "hanaSecretName": "HANA secret name",
    "prometheusPort": "Target port to emit Prometheus metrics"
}
```

Eigenschaften

- hanaSid

  Die dreistellige SAP-System-ID (SID) des SAP HANA-Systems.
- hanaPort

  Der HANA-Datenbankport, nach dem der Exporter HANA-Metriken abfragt.
- hanaSecretName

  Das AWS Secrets Manager Geheimnis, in dem die Benutzeranmeldeinformationen für die HANA-Überwachung gespeichert werden. Der HANA-Prometheus-Exporter nutzt diese Anmeldeinformationen, um eine Verbindung zur Datenbank herzustellen und HANA-Metriken abzufragen.
- prometheusPort (optional)

  Der Ziel-Port, an den Prometheus Metriken sendet. Wenn nicht angegeben, wird der Standard-Port 9668 verwendet.

HA-Cluster-Prometheus-Exporteur

Definiert die HA-Cluster-Prometheus-Exporter-Einstellungen.

JSON

```
"haClusterPrometheusExporter": {
    "prometheusPort": "Target port to emit Prometheus metrics"
```

```
}
```

Eigenschaften

- prometheusPort (optional)

  Der Ziel-Port, an den Prometheus Metriken sendet. Wenn nicht angegeben, wird der Standard-Port 9664 verwendet.

NetWeaver Prometheus-Exporteur

Definiert die NetWeaver Prometheus Exporter-Einstellungen.

JSON

```
"netWeaverPrometheusExporter": {
    "sapSid": "SAP NetWeaver  SID",
    "instanceNumbers": [ "Array of instance Numbers of SAP NetWeaver system "],
"prometheusPort": "Target port to emit Prometheus metrics"
}
```

Eigenschaften

- sapSid

  Die dreistellige SAP-System-ID (SID) des SAP-Systems. NetWeaver
- Instancenummern

  Array der Instanznummern des NetWeaver SAP-Systems.

  Beispiel: `"instanceNumbers": [ "00", "01"]`
- prometheusPort (optional)

  Der Ziel-Port, an den Prometheus Metriken senden soll. Wenn nicht angegeben, wird der Standard-Port `9680` verwendet.

SAP ASE Prometheus Exporter

Definiert die SAP-ASE-Prometheus-Exporter-Einstellungen.

JSON

```
"sapASEPrometheusExporter": {
    "sapAseSid": "SAP ASE SID",
    "sapAsePort": "SAP ASE database port",
    "sapAseSecretName": "SAP ASE secret name",
    "prometheusPort": "Target port to emit Prometheus metrics",
    "agreeToEnableASEMonitoring": true
}
```

Eigenschaften

- sapAseSid

  Die dreistellige SAP-System-ID (SID) des SAP-ASE-Systems.

- sapAsePort

  Der SAP-ASE-Datenbankport, nach dem der Exporter ASE-Metriken abfragt.

- sapAseSecretName

  Das AWS Secrets Manager Geheimnis, in dem die Benutzeranmeldeinformationen für die ASE-Überwachung gespeichert werden. Der SAP-ASE-Prometheus-Exporter nutzt diese Anmeldeinformationen, um eine Verbindung zur Datenbank herzustellen und ASE-Metriken abzufragen.

- prometheusPort (optional)

  Der Ziel-Port, an den Prometheus Metriken sendet. Wenn nicht angegeben, wird der Standard-Port 9399 verwendet. Wenn es eine andere ASE-DB gibt, die den Standardport verwendet, verwenden wir 9499.

Windows-Ereignisse

Definiert zu protokollierende Windows-Ereignisse.

JSON

```
{
  "logGroupName" : "logGroupName",
  "eventName" : "eventName",
  "eventLevels" : ["ERROR","WARNING","CRITICAL","INFORMATION","VERBOSE"],
  "monitor" : true/false
```

```
}
```

Eigenschaften

- logGroupName (erforderlich)

  Der Name der CloudWatch Protokollgruppe, die dem überwachten Protokoll zugeordnet werden soll. Informationen zu den Einschränkungen für Protokollgruppennamen finden Sie unter CreateLogGroup.
- eventName (erforderlich)

  Der Typ der zu protokollierenden Windows-Ereignisse. Dies entspricht dem Kanalnamen des Windows-Ereignisprotokolls. Zum Beispiel System CustomEventName, Sicherheit usw. Dieses Feld ist für jeden Typ eines zu protokollierenden Windows-Ereignisses ein Pflichtfeld.
- eventLevels (erforderlich)

  Die Ebenen des zu protokollierenden Ereignisses. Sie müssen jede zu protokollierende Ebene angeben. Mögliche Werte sind `INFORMATION`, `WARNING`, `ERROR`, `CRITICAL` und `VERBOSE`. Dieses Feld ist für jeden Typ eines zu protokollierenden Windows-Ereignisses ein Pflichtfeld.
- monitor (optional)

  Boolescher Wert, der angibt, ob die Protokolle überwacht werden sollen. Der Standardwert ist `true`.

Alarm

Definiert einen CloudWatch Alarm, der für die Komponente überwacht werden soll.

JSON

```
{
  "alarmName" : "monitoredAlarmName",
  "severity" : HIGH/MEDIUM/LOW
}
```

Eigenschaften

- alarmName (erforderlich)

  Der Name des CloudWatch Alarms, der für die Komponente überwacht werden soll.

- Schweregrad (optional)

  Gibt den Grad des Ausfalls an, wenn der Alarm ausgelöst wird.

Beispiele für die Komponentenkonfiguration

Die folgenden Beispiele zeigen Komponentenkonfigurationen im JSON-Format für relevante Services.

Beispiele für Komponentenkonfigurationen

- Amazon-DynamoDB-Tabelle.
- Amazon EC2 Auto Scaling (ASG)
- Amazon-EKS-Cluster
- Amazon Elastic Compute Cloud (EC2)-Instance
- Amazon Elastic Container Service (Amazon ECS)
- Amazon-ECS-Dienstleistungen
- Amazon-ECS-Aufgaben
- Amazon Elastic File System (Amazon EFS)
- Amazon FSx
- Amazon Relational Database Service (RDS) Aurora MySQL
- Amazon-Relational-Database-Service(RDS)-Instance
- Amazon-Route-53-Zustandsprüfung
- Gehostete Zone von Amazon Route 53
- Amazon Route 53 Resolver Endpunkt
- Amazon Route 53 Resolver Konfiguration der Abfrageprotokollierung
- Amazon-S3-Bucket
- Amazon Simple Queue Service (SQS)
- Amazon SNS-Thema
- Amazon Virtual Private Cloud (Amazon VPC)
- Gateways von Amazon VPC Network Address Translation (NAT)
- API-Gateway-REST-API-Phasen
- Application Elastic Load Balancing
- AWS Lambda Funktion

- AWS Network Firewall Regelgruppe
- AWS Network Firewall Regelgruppenzuweisung
- AWS Step Functions
- Vom Kunden gruppierte Amazon-EC2-Instances
- Elastic Load Balancing
- Java
- Kubernetes auf Amazon EC2
- RDS MariaDB und RDS MySQL
- RDS Oracle
- RDS PostgreSQL
- SAP ASE auf Amazon EC2
- SAP ASE bei Amazon EC2 mit hoher Verfügbarkeit
- SAP HANA bei Amazon EC2
- SAP HANA bei Amazon EC2 mit hoher Verfügbarkeit
- SAP NetWeaver auf Amazon EC2
- NetWeaver SAP-Hochverfügbarkeit auf Amazon EC2
- SQL Always On-Verfügbarkeitsgruppe
- SQL-Failoverclusterinstance

Amazon-DynamoDB-Tabelle.

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für eine Amazon-DynamoDB-Tabelle.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "SystemErrors",
      "monitor": false
    },
    {
      "alarmMetricName": "UserErrors",
      "monitor": false
    },
    {
```

```
      "alarmMetricName": "ConsumedReadCapacityUnits",
      "monitor": false
    },
    {
      "alarmMetricName": "ConsumedWriteCapacityUnits",
      "monitor": false
    },
    {
      "alarmMetricName": "ReadThrottleEvents",
      "monitor": false
    },
    {
      "alarmMetricName": "WriteThrottleEvents",
      "monitor": false
    },
    {
      "alarmMetricName": "ConditionalCheckFailedRequests",
      "monitor": false
    },
    {
      "alarmMetricName": "TransactionConflict",
      "monitor": false
    }
  ],
  "logs": []
}
```

Amazon EC2 Auto Scaling (ASG)

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für Amazon EC2 Auto Scaling (ASG).

```
{
    "alarmMetrics" : [
      {
        "alarmMetricName" : "CPUCreditBalance"
      }, {
        "alarmMetricName" : "EBSIOBalance%"
      }
    ],
    "subComponents" : [
      {
        "subComponentType" : "AWS::EC2::Instance",
```

```
        "alarmMetrics" : [
          {
            "alarmMetricName" : "CPUUtilization"
          }, {
            "alarmMetricName" : "StatusCheckFailed"
          }
        ],
        "logs" : [
          {
            "logGroupName" : "my_log_group",
            "logPath" : "C:\\LogFolder\\*",
            "logType" : "APPLICATION"
          }
        ],
        "processes" : [
          {
            "processName" : "my_process",
            "alarmMetrics" : [
              {
                  "alarmMetricName" : "procstat cpu_usage",
                  "monitor" : true
              }, {
                  "alarmMetricName" : "procstat memory_rss",
                  "monitor" : true
              }
            ]
      }
  ],
        "windowsEvents" : [
          {
            "logGroupName" : "my_log_group_2",
            "eventName" : "Application",
            "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ]
          }
        ]
      }, {
        "subComponentType" : "AWS::EC2::Volume",
        "alarmMetrics" : [
          {
            "alarmMetricName" : "VolumeQueueLength"
          }, {
            "alarmMetricName" : "BurstBalance"
          }
        ]
```

```
      }
    ],
    "alarms" : [
      {
        "alarmName" : "my_asg_alarm",
        "severity" : "LOW"
      }
    ]
  }
```

Amazon-EKS-Cluster

Das folgende Beispiel zeigt Komponentenkonfigurationen im JSON-Format für den Amazon-EKS-Cluster.

```
{
    "alarmMetrics":[
       {
          "alarmMetricName": "cluster_failed_node_count",
          "monitor":true
       },
       {
          "alarmMetricName": "node_cpu_reserved_capacity",
          "monitor":true
       },
       {
          "alarmMetricName": "node_cpu_utilization",
          "monitor":true
       },
       {
          "alarmMetricName": "node_filesystem_utilization",
          "monitor":true
       },
       {
          "alarmMetricName": "node_memory_reserved_capacity",
          "monitor":true
       },
       {
          "alarmMetricName": "node_memory_utilization",
          "monitor":true
       },
       {
          "alarmMetricName": "node_network_total_bytes",
```

```
            "monitor":true
         },
         {
            "alarmMetricName": "pod_cpu_reserved_capacity",
            "monitor":true
         },
         {
            "alarmMetricName": "pod_cpu_utilization",
            "monitor":true
         },
         {
            "alarmMetricName": "pod_cpu_utilization_over_pod_limit",
            "monitor":true
         },
         {
            "alarmMetricName": "pod_memory_reserved_capacity",
            "monitor":true
         },
         {
            "alarmMetricName": "pod_memory_utilization",
            "monitor":true
         },
         {
            "alarmMetricName": "pod_memory_utilization_over_pod_limit",
            "monitor":true
         },
         {
            "alarmMetricName": "pod_network_rx_bytes",
            "monitor":true
         },
         {
            "alarmMetricName": "pod_network_tx_bytes",
            "monitor":true
         }
      ],
      "logs":[
         {
            "logGroupName": "/aws/containerinsights/kubernetes/application",
            "logType":"APPLICATION",
            "monitor":true,
            "encoding":"utf-8"
         }
      ],
      "subComponents":[
```

```
        {
           "subComponentType":"AWS::EC2::Instance",
           "alarmMetrics":[
              {
                 "alarmMetricName":"CPUUtilization",
                 "monitor":true
              },
              {
                 "alarmMetricName":"StatusCheckFailed",
                 "monitor":true
              },
              {
                 "alarmMetricName":"disk_used_percent",
                 "monitor":true
              },
              {
                 "alarmMetricName":"mem_used_percent",
                 "monitor":true
              }
           ],
           "logs":[
              {
                 "logGroupName":"APPLICATION-KubernetesClusterOnEC2-IAD",
                 "logPath":"",
                 "logType":"APPLICATION",
                 "monitor":true,
                 "encoding":"utf-8"
              }
           ],
           "processes" : [
             {
                 "processName" : "my_process",
                 "alarmMetrics" : [
                     {
                         "alarmMetricName" : "procstat cpu_usage",
                         "monitor" : true
                     }, {
                         "alarmMetricName" : "procstat memory_rss",
                         "monitor" : true
                     }
                 ]
             }
         ],
           "windowsEvents":[
```

```
            {
               "logGroupName":"my_log_group_2",
               "eventName":"Application",
               "eventLevels":[
                  "ERROR",
                  "WARNING",
                  "CRITICAL"
               ],
               "monitor":true
            }
         ]
      },
      {
         "subComponentType":"AWS::AutoScaling::AutoScalingGroup",
         "alarmMetrics":[
            {
               "alarmMetricName":"CPUCreditBalance",
               "monitor":true
            },
            {
               "alarmMetricName":"EBSIOBalance%",
               "monitor":true
            }
         ]
      },
      {
         "subComponentType":"AWS::EC2::Volume",
         "alarmMetrics":[
            {
               "alarmMetricName":"VolumeReadBytes",
               "monitor":true
            },
            {
               "alarmMetricName":"VolumeWriteBytes",
               "monitor":true
            },
            {
               "alarmMetricName":"VolumeReadOps",
               "monitor":true
            },
            {
               "alarmMetricName":"VolumeWriteOps",
               "monitor":true
            },
```

```
            {
               "alarmMetricName":"VolumeQueueLength",
               "monitor":true
            },
            {
               "alarmMetricName":"BurstBalance",
               "monitor":true
            }
         ]
      }
   ]
}
```

> **Note**
>
> - Der Abschnitt `subComponents` von `AWS::EC2::Instance`, `AWS::EC2::Volume` und `AWS::AutoScaling::AutoScalingGroup` gilt nur für Amazon-EKS-Cluster, die auf dem EC2-Starttyp ausgeführt werden.
> - Der Abschnitt `windowsEvents` von `AWS::EC2::Instance` in `subComponents` gilt nur für Windows, das auf Amazon-EC2-Instances ausgeführt wird.

Amazon Elastic Compute Cloud (EC2)-Instance

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für eine Amazon-EC2-Instance.

> **Important**
>
> Wenn eine Amazon-EC2-Instance in den Status `stopped` wechselt, wird sie aus der Überwachung entfernt. Wenn es in einen `running` Status zurückkehrt, wird es der Liste der nicht überwachten Komponenten auf der Seite mit den Anwendungsdetails der CloudWatch Application Insights-Konsole hinzugefügt. Wenn die automatische Überwachung neuer Ressourcen für die Anwendung aktiviert ist, wird die Instance zur Liste der Überwachten Komponenten hinzugefügt. Die Protokolle und Metriken werden jedoch auf den Standardwert für die Workload festgelegt. Die vorherige Protokoll- und Metrikkonfiguration wird nicht gespeichert.

```
{
    "alarmMetrics" : [
      {
        "alarmMetricName" : "CPUUtilization",
        "monitor" : true
      }, {
        "alarmMetricName" : "StatusCheckFailed"
      }
    ],
    "logs" : [
      {
        "logGroupName" : "my_log_group",
        "logPath" : "C:\\LogFolder\\*",
        "logType" : "APPLICATION",
        "monitor" : true
      },
      {
        "logGroupName" : "my_log_group_2",
        "logPath" : "C:\\LogFolder2\\*",
        "logType" : "IIS",
        "encoding" : "utf-8"
      }
    ],
    "processes" : [
        {
            "processName" : "my_process",
            "alarmMetrics" : [
                {
                    "alarmMetricName" : "procstat cpu_usage",
                    "monitor" : true
                }, {
                    "alarmMetricName" : "procstat memory_rss",
                    "monitor" : true
                }
            ]
        }
    ],
    "windowsEvents" : [
      {
        "logGroupName" : "my_log_group_3",
        "eventName" : "Application",
        "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ],
        "monitor" : true
```

```
      }, {
        "logGroupName" : "my_log_group_4",
        "eventName" : "System",
        "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ],
        "monitor" : true
    }],
     "alarms" : [
      {
        "alarmName" : "my_instance_alarm_1",
        "severity" : "HIGH"
      },
      {
        "alarmName" : "my_instance_alarm_2",
        "severity" : "LOW"
      }
    ],
     "subComponents" : [
     {
       "subComponentType" : "AWS::EC2::Volume",
       "alarmMetrics" : [
       {
         "alarmMetricName" : "VolumeQueueLength",
         "monitor" : "true"
       },
       {
         "alarmMetricName" : "VolumeThroughputPercentage",
         "monitor" : "true"
       },
       {
         "alarmMetricName" : "BurstBalance",
         "monitor" : "true"
       }
    }]
  }
```

Amazon Elastic Container Service (Amazon ECS)

Das folgende Beispiel zeigt Komponentenkonfigurationen im JSON-Format für den Amazon Elastic Container Service (Amazon ECS).

```
{
    "alarmMetrics":[
       {
```

```
            "alarmMetricName":"CpuUtilized",
            "monitor":true
         },
         {
            "alarmMetricName":"MemoryUtilized",
            "monitor":true
         },
         {
            "alarmMetricName":"NetworkRxBytes",
            "monitor":true
         },
         {
            "alarmMetricName":"NetworkTxBytes",
            "monitor":true
         },
         {
            "alarmMetricName":"RunningTaskCount",
            "monitor":true
         },
         {
            "alarmMetricName":"PendingTaskCount",
            "monitor":true
         },
         {
            "alarmMetricName":"StorageReadBytes",
            "monitor":true
         },
         {
            "alarmMetricName":"StorageWriteBytes",
            "monitor":true
         }
      ],
      "logs":[
         {
            "logGroupName":"/ecs/my-task-definition",
            "logType":"APPLICATION",
            "monitor":true
         }
      ],
      "subComponents":[
         {
            "subComponentType":"AWS::ElasticLoadBalancing::LoadBalancer",
            "alarmMetrics":[
               {
```

```
                "alarmMetricName":"HTTPCode_Backend_4XX",
                "monitor":true
             },
             {
                "alarmMetricName":"HTTPCode_Backend_5XX",
                "monitor":true
             },
             {
                "alarmMetricName":"Latency",
                "monitor":true
             },
             {
                "alarmMetricName":"SurgeQueueLength",
                "monitor":true
             },
             {
                "alarmMetricName":"UnHealthyHostCount",
                "monitor":true
             }
          ]
       },
       {
          "subComponentType":"AWS::ElasticLoadBalancingV2::LoadBalancer",
          "alarmMetrics":[
             {
                "alarmMetricName":"HTTPCode_Target_4XX_Count",
                "monitor":true
             },
             {
                "alarmMetricName":"HTTPCode_Target_5XX_Count",
                "monitor":true
             },
             {
                "alarmMetricName":"TargetResponseTime",
                "monitor":true
             },
             {
                "alarmMetricName":"UnHealthyHostCount",
                "monitor":true
             }
          ]
       },
       {
          "subComponentType":"AWS::EC2::Instance",
```

```
        "alarmMetrics":[
           {
              "alarmMetricName":"CPUUtilization",
              "monitor":true
           },
           {
              "alarmMetricName":"StatusCheckFailed",
              "monitor":true
           },
           {
              "alarmMetricName":"disk_used_percent",
              "monitor":true
           },
           {
              "alarmMetricName":"mem_used_percent",
              "monitor":true
           }
        ],
        "logs":[
           {
              "logGroupName":"my_log_group",
              "logPath":"/mylog/path",
              "logType":"APPLICATION",
              "monitor":true
           }
        ],
        "processes" : [
           {
              "processName" : "my_process",
              "alarmMetrics" : [
                  {
                      "alarmMetricName" : "procstat cpu_usage",
                      "monitor" : true
                  }, {
                      "alarmMetricName" : "procstat memory_rss",
                      "monitor" : true
                  }
               ]
           }
        ],
        "windowsEvents":[
           {
              "logGroupName":"my_log_group_2",
              "eventName":"Application",
```

```
                "eventLevels":[
                   "ERROR",
                   "WARNING",
                   "CRITICAL"
                ],
                "monitor":true
             }
          ]
       },
       {
          "subComponentType":"AWS::EC2::Volume",
          "alarmMetrics":[
             {
                "alarmMetricName":"VolumeQueueLength",
                "monitor":"true"
             },
             {
                "alarmMetricName":"VolumeThroughputPercentage",
                "monitor":"true"
             },
             {
                "alarmMetricName":"BurstBalance",
                "monitor":"true"
             }
          ]
       }
    ]
}
```

> **Note**
>
> - Der Abschnitt `subComponents` von `AWS::EC2::Instance` und `AWS::EC2::Volume` gilt nur für Amazon-ECS-Cluster mit ECS-Service oder ECS-Aufgabe, die auf dem EC2-Starttyp ausgeführt werden.
> - Der Abschnitt `windowsEvents` von `AWS::EC2::Instance` in `subComponents` gilt nur für Windows, das auf Amazon-EC2-Instances ausgeführt wird.

Amazon-ECS-Dienstleistungen

Das folgende Beispiel zeigt Komponentenkonfigurationen im JSON-Format für einen Amazon-ECS-Service.

```
{
    "alarmMetrics":[
       {
          "alarmMetricName":"CPUUtilization",
          "monitor":true
       },
       {
          "alarmMetricName":"MemoryUtilization",
          "monitor":true
       },
       {
          "alarmMetricName":"CpuUtilized",
          "monitor":true
       },
       {
          "alarmMetricName":"MemoryUtilized",
          "monitor":true
       },
       {
          "alarmMetricName":"NetworkRxBytes",
          "monitor":true
       },
       {
          "alarmMetricName":"NetworkTxBytes",
          "monitor":true
       },
       {
          "alarmMetricName":"RunningTaskCount",
          "monitor":true
       },
       {
          "alarmMetricName":"PendingTaskCount",
          "monitor":true
       },
       {
          "alarmMetricName":"StorageReadBytes",
          "monitor":true
       },
       {
```

```
            "alarmMetricName":"StorageWriteBytes",
            "monitor":true
         }
      ],
      "logs":[
         {
            "logGroupName":"/ecs/my-task-definition",
            "logType":"APPLICATION",
            "monitor":true
         }
      ],
      "subComponents":[
         {
            "subComponentType":"AWS::ElasticLoadBalancing::LoadBalancer",
            "alarmMetrics":[
               {
                  "alarmMetricName":"HTTPCode_Backend_4XX",
                  "monitor":true
               },
               {
                  "alarmMetricName":"HTTPCode_Backend_5XX",
                  "monitor":true
               },
               {
                  "alarmMetricName":"Latency",
                  "monitor":true
               },
               {
                  "alarmMetricName":"SurgeQueueLength",
                  "monitor":true
               },
               {
                  "alarmMetricName":"UnHealthyHostCount",
                  "monitor":true
               }
            ]
         },
         {
            "subComponentType":"AWS::ElasticLoadBalancingV2::LoadBalancer",
            "alarmMetrics":[
               {
                  "alarmMetricName":"HTTPCode_Target_4XX_Count",
                  "monitor":true
               },
```

```
            {
               "alarmMetricName":"HTTPCode_Target_5XX_Count",
               "monitor":true
            },
            {
               "alarmMetricName":"TargetResponseTime",
               "monitor":true
            },
            {
               "alarmMetricName":"UnHealthyHostCount",
               "monitor":true
            }
         ]
      },
      {
         "subComponentType":"AWS::EC2::Instance",
         "alarmMetrics":[
            {
               "alarmMetricName":"CPUUtilization",
               "monitor":true
            },
            {
               "alarmMetricName":"StatusCheckFailed",
               "monitor":true
            },
            {
               "alarmMetricName":"disk_used_percent",
               "monitor":true
            },
            {
               "alarmMetricName":"mem_used_percent",
               "monitor":true
            }
         ],
         "logs":[
            {
               "logGroupName":"my_log_group",
               "logPath":"/mylog/path",
               "logType":"APPLICATION",
               "monitor":true
            }
         ],
         "processes" : [
            {
```

```
                "processName" : "my_process",
                "alarmMetrics" : [
                {
                    "alarmMetricName" : "procstat cpu_usage",
                    "monitor" : true
                }, {
                    "alarmMetricName" : "procstat memory_rss",
                    "monitor" : true
                }
            ]
        }
    ],
          "windowsEvents":[
             {
                "logGroupName":"my_log_group_2",
                "eventName":"Application",
                "eventLevels":[
                   "ERROR",
                   "WARNING",
                   "CRITICAL"
                ],
                "monitor":true
             }
          ]
       },
       {
          "subComponentType":"AWS::EC2::Volume",
          "alarmMetrics":[
             {
                "alarmMetricName":"VolumeQueueLength",
                "monitor":"true"
             },
             {
                "alarmMetricName":"VolumeThroughputPercentage",
                "monitor":"true"
             },
             {
                "alarmMetricName":"BurstBalance",
                "monitor":"true"
             }
          ]
       }
    ]
```

```
}
```

> **Note**
>
> - Der Abschnitt `subComponents` von `AWS::EC2::Instance` und `AWS::EC2::Volume` gilt nur für Amazon ECS, das auf dem EC2-Starttyp ausgeführt wird.
> - Der Abschnitt `windowsEvents` von `AWS::EC2::Instance` in `subComponents` gilt nur für Windows, das auf Amazon-EC2-Instances ausgeführt wird.

Amazon-ECS-Aufgaben

Das folgende Beispiel zeigt Komponentenkonfigurationen im JSON-Format für eine Amazon-ECS-Aufgabe.

```
{
    "logs":[
       {
          "logGroupName":"/ecs/my-task-definition",
          "logType":"APPLICATION",
          "monitor":true
       }
    ],
    "processes" : [
        {
            "processName" : "my_process",
            "alarmMetrics" : [
                {
                    "alarmMetricName" : "procstat cpu_usage",
                    "monitor" : true
                }, {
                    "alarmMetricName" : "procstat memory_rss",
                    "monitor" : true
                }
            ]
        }
    ]
}
```

Amazon Elastic File System (Amazon EFS)

Das folgende Beispiel zeigt Komponentenkonfigurationen im JSON-Format für die Amazon EFS.

```
{
   "alarmMetrics": [
     {
       "alarmMetricName": "BurstCreditBalance",
       "monitor": true
     },
     {
       "alarmMetricName": "PercentIOLimit",
       "monitor": true
     },
     {
       "alarmMetricName": "PermittedThroughput",
       "monitor": true
     },
     {
       "alarmMetricName": "MeteredIOBytes",
       "monitor": true
     },
     {
       "alarmMetricName": "TotalIOBytes",
       "monitor": true
     },
     {
       "alarmMetricName": "DataWriteIOBytes",
       "monitor": true
     },
     {
       "alarmMetricName": "DataReadIOBytes",
       "monitor": true
     },
     {
       "alarmMetricName": "MetadataIOBytes",
       "monitor": true
     },
     {
       "alarmMetricName": "ClientConnections",
       "monitor": true
     },
     {
       "alarmMetricName": "TimeSinceLastSync",
```

```
      "monitor": true
    },
    {
      "alarmMetricName": "Throughput",
      "monitor": true
    },
    {
      "alarmMetricName": "PercentageOfPermittedThroughputUtilization",
      "monitor": true
    },
    {
      "alarmMetricName": "ThroughputIOPS",
      "monitor": true
    },
    {
      "alarmMetricName": "PercentThroughputDataReadIOBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "PercentThroughputDataWriteIOBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "PercentageOfIOPSDataReadIOBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "PercentageOfIOPSDataWriteIOBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "AverageDataReadIOBytesSize",
      "monitor": true
    },
    {
      "alarmMetricName": "AverageDataWriteIOBytesSize",
      "monitor": true
    }
  ],
  "logs": [
   {
   "logGroupName": "/aws/efs/utils",
   "logType": "EFS_MOUNT_STATUS",
   "monitor": true,
```

```
    }
   ]
 }
```

Amazon FSx

Das folgende Beispiel zeigt Komponentenkonfigurationen im JSON-Format für die Amazon FSx.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "DataReadBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "DataWriteBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "DataReadOperations",
      "monitor": true
    },
    {
      "alarmMetricName": "DataWriteOperations",
      "monitor": true
    },
    {
      "alarmMetricName": "MetadataOperations",
      "monitor": true
    },
    {
      "alarmMetricName": "FreeStorageCapacity",
      "monitor": true
    }
  ]
}
```

Amazon Relational Database Service (RDS) Aurora MySQL

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für Amazon RDS Aurora MySQL.

```
{
```

```
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    },
    {
      "alarmMetricName": "CommitLatency",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "MYSQL",
      "monitor": true,
    },
    {
      "logType": "MYSQL_SLOW_QUERY",
      "monitor": false
    }
  ]
}
```

Amazon-Relational-Database-Service(RDS)-Instance

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für eine Amazon-RDS-Instance.

```
{
  "alarmMetrics" : [
    {
      "alarmMetricName" : "BurstBalance",
      "monitor" : true
    }, {
      "alarmMetricName" : "WriteThroughput",
      "monitor" : false
    }
  ],

  "alarms" : [
    {
      "alarmName" : "my_rds_instance_alarm",
      "severity" : "MEDIUM"
    }
```

```
  ]
}
```

Amazon-Route-53-Zustandsprüfung

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für die Zustandsprüfung von Amazon Route 53.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "ChildHealthCheckHealthyCount",
      "monitor": true
    },
    {
      "alarmMetricName": "ConnectionTime",
      "monitor": true
    },
    {
      "alarmMetricName": "HealthCheckPercentageHealthy",
      "monitor": true
    },
    {
      "alarmMetricName": "HealthCheckStatus",
      "monitor": true
    },
    {
      "alarmMetricName": "SSLHandshakeTime",
      "monitor": true
    },
    {
      "alarmMetricName": "TimeToFirstByte",
      "monitor": true
    }
  ]
}
```

Gehostete Zone von Amazon Route 53

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für eine von Amazon Route 53 gehostete Zone.

```
{
```

```
  "alarmMetrics": [
    {
      "alarmMetricName": "DNSQueries",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECInternalFailure",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECKeySigningKeysNeedingAction",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECKeySigningKeyMaxNeedingActionAge",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECKeySigningKeyAge",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logGroupName":"/hosted-zone/logs",
      "logType": "ROUTE53_DNS_PUBLIC_QUERY_LOGS",
      "monitor": true
    }
  ]
}
```

Amazon Route 53 Resolver Endpunkt

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für Amazon Route 53 Resolver Endgeräte.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "EndpointHealthyENICount",
      "monitor": true
    },
    {
```

```
      "alarmMetricName": "EndpointUnHealthyENICount",
      "monitor": true
    },
    {
      "alarmMetricName": "InboundQueryVolume",
      "monitor": true
    },
    {
      "alarmMetricName": "OutboundQueryVolume",
      "monitor": true
    },
    {
      "alarmMetricName": "OutboundQueryAggregateVolume",
      "monitor": true
    }
  ]
}
```

Amazon Route 53 Resolver Konfiguration der Abfrageprotokollierung

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für die Amazon Route 53 Resolver Konfiguration zur Abfrageprotokollierung.

```
{
  "logs": [
    {
      "logGroupName": "/resolver-query-log-config/logs",
      "logType": "ROUTE53_RESOLVER_QUERY_LOGS",
      "monitor": true
    }
  ]
}
```

Amazon-S3-Bucket

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für ein Amazon S3 Bucket.

```
{
    "alarmMetrics" : [
        {
            "alarmMetricName" : "ReplicationLatency",
            "monitor" : true
```

```
        }, {
            "alarmMetricName" : "5xxErrors",
            "monitor" : true
        }, {
            "alarmMetricName" : "BytesDownloaded"
            "monitor" : true
        }
    ]
}
```

Amazon Simple Queue Service (SQS)

Die folgenden Beispiele zeigen Komponentenkonfigurationen im JSON-Format für den Amazon Simple Queue Service.

```
{
  "alarmMetrics" : [
    {
      "alarmMetricName" : "ApproximateAgeOfOldestMessage"
    }, {
      "alarmMetricName" : "NumberOfEmptyReceives"
    }
  ],
  "alarms" : [
    {
      "alarmName" : "my_sqs_alarm",
      "severity" : "MEDIUM"
    }
  ]
}
```

Amazon SNS-Thema

Das folgende Beispiel zeigt Komponentenkonfigurationen im JSON-Format für das Amazon-SNS-Thema.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "NumberOfNotificationsFailed",
      "monitor": true
    },
    {
```

```
      "alarmMetricName": "NumberOfNotificationsFilteredOut-InvalidAttributes",
      "monitor": true
    },
    {
      "alarmMetricName": "NumberOfNotificationsFilteredOut-NoMessageAttributes",
      "monitor": true
    },
    {
      "alarmMetricName": "NumberOfNotificationsFailedToRedriveToDlq",
      "monitor": true
    }
  ]
}
```

Amazon Virtual Private Cloud (Amazon VPC)

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für Amazon VPC.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "NetworkAddressUsage",
      "monitor": true
    },
    {
      "alarmMetricName": "NetworkAddressUsagePeered",
      "monitor": true
    },
    {
      "alarmMetricName": "VPCFirewallQueryVolume",
      "monitor": true
    }
  ]
}
```

Gateways von Amazon VPC Network Address Translation (NAT)

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für NAT-Gateways.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "ErrorPortAllocation",
```

```
      "monitor": true
    },
    {
      "alarmMetricName": "IdleTimeoutCount",
      "monitor": true
    }
  ]
}
```

API-Gateway-REST-API-Phasen

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für API-Gateway-REST-API-Phasen.

```
{
     "alarmMetrics" : [
         {
             "alarmMetricName" : "4XXError",
             "monitor" : true
         },
         {
             "alarmMetricName" : "5XXError",
             "monitor" : true
         }
     ],
    "logs" : [
        {
            "logType" : "API_GATEWAY_EXECUTION",
            "monitor" : true
        },
        {
            "logType" : "API_GATEWAY_ACCESS",
            "monitor" : true
        }
    ]
}
```

Application Elastic Load Balancing

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für Application Elastic Load Balancing.

```
{
```

```
  "alarmMetrics": [
    {
      "alarmMetricName": "ActiveConnectionCount",
    }, {
      "alarmMetricName": "TargetResponseTime"
    }
  ],
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "CPUUtilization",
        }, {
          "alarmMetricName": "StatusCheckFailed"
        }
      ],
      "logs": [
        {
          "logGroupName": "my_log_group",
          "logPath": "C:\\LogFolder\\*",
          "logType": "APPLICATION",
        }
      ],
      "windowsEvents": [
        {
          "logGroupName": "my_log_group_2",
          "eventName": "Application",
          "eventLevels": [ "ERROR", "WARNING", "CRITICAL" ]
        }
      ]
    }, {
      "subComponentType": "AWS::EC2::Volume",
      "alarmMetrics": [
        {
          "alarmMetricName": "VolumeQueueLength",
        }, {
          "alarmMetricName": "BurstBalance"
        }
      ]
    }
  ],

  "alarms": [
```

```
    {
      "alarmName": "my_alb_alarm",
      "severity": "LOW"
    }
  ]
}
```

AWS Lambda Funktion

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für eine AWS Lambda -Funktion.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "Errors",
      "monitor": true
    },
    {
      "alarmMetricName": "Throttles",
      "monitor": true
    },
    {
      "alarmMetricName": "IteratorAge",
      "monitor": true
    },
    {
      "alarmMetricName": "Duration",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "DEFAULT",
      "monitor": true
    }
  ]
}
```

AWS Network Firewall Regelgruppe

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für eine AWS Network Firewall -Regelgruppe.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "FirewallRuleGroupQueryVolume",
      "monitor": true
    }
  ]
}
```

AWS Network Firewall Regelgruppenzuweisung

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für eine AWS Network Firewall -Regelgruppenzuordnung.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "FirewallRuleGroupQueryVolume",
      "monitor": true
    }
  ]
}
```

AWS Step Functions

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für AWS Step Functions.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "ExecutionsFailed",
      "monitor": true
    },
    {
      "alarmMetricName": "LambdaFunctionsFailed",
      "monitor": true
    },
    {
      "alarmMetricName": "ProvisionedRefillRate",
      "monitor": true
    }
```

```
  ],
  "logs": [
    {
     "logGroupName": "/aws/states/HelloWorld-Logs",
      "logType": "STEP_FUNCTION",
      "monitor": true,
    }
  ]
}
```

Vom Kunden gruppierte Amazon-EC2-Instances

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für eine vom Kunden gruppierte Amazon-EC2-Instance.

```
{
    "subComponents": [
      {
        "subComponentType": "AWS::EC2::Instance",
        "alarmMetrics": [
          {
            "alarmMetricName": "CPUUtilization",
          },
          {
            "alarmMetricName": "StatusCheckFailed"
          }
        ],
        "logs": [
          {
            "logGroupName": "my_log_group",
            "logPath": "C:\\LogFolder\\*",
            "logType": "APPLICATION",
          }
        ],
        "processes": [
            {
                "processName": "my_process",
                "alarmMetrics": [
                    {
                        "alarmMetricName": "procstat cpu_usage",
                        "monitor": true
                    }, {
                        "alarmMetricName": "procstat memory_rss",
```

```
                        "monitor": true
                    }
                ]
            }
        ],
        "windowsEvents": [
          {
            "logGroupName": "my_log_group_2",
            "eventName": "Application",
            "eventLevels": [ "ERROR", "WARNING", "CRITICAL" ]
          }
        ]
      }, {
        "subComponentType": "AWS::EC2::Volume",
        "alarmMetrics": [
          {
            "alarmMetricName": "VolumeQueueLength",
          }, {
            "alarmMetricName": "BurstBalance"
          }
        ]
      }
    ],
    "alarms": [
      {
        "alarmName": "my_alarm",
        "severity": "MEDIUM"
      }
    ]
  }
```

Elastic Load Balancing

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für Elastic Load Balancing.

```
{
    "alarmMetrics": [
      {
        "alarmMetricName": "EstimatedALBActiveConnectionCount"
      }, {
        "alarmMetricName": "HTTPCode_Backend_5XX"
      }
```

```
    ],
    "subComponents": [
      {
        "subComponentType": "AWS::EC2::Instance",
        "alarmMetrics": [
          {
            "alarmMetricName": "CPUUtilization"
          }, {
            "alarmMetricName": "StatusCheckFailed"
          }
        ],
        "logs": [
          {
            "logGroupName": "my_log_group",
            "logPath": "C:\\LogFolder\\*",
            "logType": "APPLICATION"
          }
        ],
        "processes": [
          {
            "processName": "my_process",
            "alarmMetrics": [
                {
                  "alarmMetricName": "procstat cpu_usage",
                  "monitor": true
                }, {
                  "alarmMetricName": "procstat memory_rss",
                  "monitor": true
                }
              ]
          }
        ],
        "windowsEvents": [
          {
            "logGroupName": "my_log_group_2",
            "eventName": "Application",
            "eventLevels": [ "ERROR", "WARNING", "CRITICAL" ],
            "monitor": true
          }
        ]
      }, {
        "subComponentType": "AWS::EC2::Volume",
        "alarmMetrics": [
          {
```

```
            "alarmMetricName": "VolumeQueueLength"
          }, {
            "alarmMetricName": "BurstBalance"
          }
        ]
      }
    ],

    "alarms": [
      {
        "alarmName": "my_elb_alarm",
        "severity": "HIGH"
      }
    ]
  }
```

Java

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für Java.

```
{
  "alarmMetrics": [ {
    "alarmMetricName": "java_lang_threading_threadcount",
    "monitor": true
  },
  {
    "alarmMetricName": "java_lang_memory_heapmemoryusage_used",
    "monitor": true
  },
  {
    "alarmMetricName": "java_lang_memory_heapmemoryusage_committed",
    "monitor": true
  }],
  "logs": [ ],
  "JMXPrometheusExporter": {
      "hostPort": "8686",
      "prometheusPort": "9404"
  }
}
```

> **Note**
>
> Application Insights unterstützt die Konfiguration der Authentifizierung für Prometheus JMX Exporter nicht. Informationen zum Einrichten der Authentifizierung finden Sie in der Beispielkonfiguration des Prometheus-JMX-Exporters.

Kubernetes auf Amazon EC2

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für Kubernetes auf Amazon EC2.

```
{
    "alarmMetrics":[
       {
          "alarmMetricName":"cluster_failed_node_count",
          "monitor":true
       },
       {
          "alarmMetricName":"node_cpu_reserved_capacity",
          "monitor":true
       },
       {
          "alarmMetricName":"node_cpu_utilization",
          "monitor":true
       },
       {
          "alarmMetricName":"node_filesystem_utilization",
          "monitor":true
       },
       {
          "alarmMetricName":"node_memory_reserved_capacity",
          "monitor":true
       },
       {
          "alarmMetricName":"node_memory_utilization",
          "monitor":true
       },
       {
          "alarmMetricName":"node_network_total_bytes",
          "monitor":true
       },
```

```
        {
           "alarmMetricName":"pod_cpu_reserved_capacity",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_cpu_utilization",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_cpu_utilization_over_pod_limit",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_memory_reserved_capacity",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_memory_utilization",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_memory_utilization_over_pod_limit",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_network_rx_bytes",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_network_tx_bytes",
           "monitor":true
        }
     ],
     "logs":[
        {
           "logGroupName":"/aws/containerinsights/kubernetes/application",
           "logType":"APPLICATION",
           "monitor":true,
           "encoding":"utf-8"
        }
     ],
     "subComponents":[
        {
           "subComponentType":"AWS::EC2::Instance",
```

```
          "alarmMetrics":[
             {
                "alarmMetricName":"CPUUtilization",
                "monitor":true
             },
             {
                "alarmMetricName":"StatusCheckFailed",
                "monitor":true
             },
             {
                "alarmMetricName":"disk_used_percent",
                "monitor":true
             },
             {
                "alarmMetricName":"mem_used_percent",
                "monitor":true
             }
          ],
          "logs":[
             {
                "logGroupName":"APPLICATION-KubernetesClusterOnEC2-IAD",
                "logPath":"",
                "logType":"APPLICATION",
                "monitor":true,
                "encoding":"utf-8"
             }
          ],
          "processes" : [
            {
                "processName" : "my_process",
                "alarmMetrics" : [
                    {
                        "alarmMetricName" : "procstat cpu_usage",
                        "monitor" : true
                    }, {
                        "alarmMetricName" : "procstat memory_rss",
                        "monitor" : true
                    }
                ]
            }
       ]
      },
      {
         "subComponentType":"AWS::EC2::Volume",
```

```
          "alarmMetrics":[
             {
                "alarmMetricName":"VolumeReadBytes",
                "monitor":true
             },
             {
                "alarmMetricName":"VolumeWriteBytes",
                "monitor":true
             },
             {
                "alarmMetricName":"VolumeReadOps",
                "monitor":true
             },
             {
                "alarmMetricName":"VolumeWriteOps",
                "monitor":true
             },
             {
                "alarmMetricName":"VolumeQueueLength",
                "monitor":true
             },
             {
                "alarmMetricName":"BurstBalance",
                "monitor":true
             }
          ]
       }
    ]
 }
```

RDS MariaDB und RDS MySQL

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für RDS MariaDB und RDS MySQL.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    }
  ],
  "logs": [
```

```
    {
      "logType": "MYSQL",
      "monitor": true,
    },
    {
      "logType": "MYSQL_SLOW_QUERY",
      "monitor": false
    }
  ]
}
```

RDS Oracle

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für RDS Oracle.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "ORACLE_ALERT",
      "monitor": true,
    },
    {
      "logType": "ORACLE_LISTENER",
      "monitor": false
    }
  ]
}
```

RDS PostgreSQL

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für RDS PostgreSQL.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
```

```
    }
  ],
  "logs": [
    {
      "logType": "POSTGRESQL",
      "monitor": true
    }
  ]
}
```

SAP ASE auf Amazon EC2

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für SAP ASE auf Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "asedb_database_availability",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_trunc_log_on_chkpt_enabled",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_last_db_backup_age_in_days",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_last_transaction_log_backup_age_in_hours",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_suspected_database",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_db_space_usage_percent",
          "monitor": true
```

```
        },
        {
          "alarmMetricName": "asedb_db_log_space_usage_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_locked_login",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_data_cache_hit_ratio",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_ASE_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2.log",
          "logType": "SAP_ASE_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_BACKUP_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2_BS.log",
          "logType": "SAP_ASE_BACKUP_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
  ],
  "sapAsePrometheusExporter": {
    "sapAseSid": "ASE",
    "sapAsePort": "4901",
    "sapAseSecretName": "ASE_DB_CREDS",
    "prometheusPort": "9399",
    "agreeToEnableASEMonitoring": true
}
```

SAP ASE bei Amazon EC2 mit hoher Verfügbarkeit

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für SAP ASE auf Amazon EC2 mit hoher Verfügbarkeit.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "asedb_database_availability",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_trunc_log_on_chkpt_enabled",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_last_db_backup_age_in_days",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_last_transaction_log_backup_age_in_hours",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_suspected_database",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_db_space_usage_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_ha_replication_state",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_ha_replication_mode",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_ha_replication_latency_in_minutes",
          "monitor": true
        }
      ],
      "logs": [
```

```
        {
          "logGroupName": "SAP_ASE_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2.log",
          "logType": "SAP_ASE_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_BACKUP_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2_BS.log",
          "logType": "SAP_ASE_BACKUP_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_REP_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/DM/repservername/repservername.log",
          "logType": "SAP_ASE_REP_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_RMA_AGENT_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/DM/RMA-*/instances/AgentContainer/logs/",
          "logType": "SAP_ASE_RMA_AGENT_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_FAULT_MANAGER_LOGS-my-resource-group",
          "logPath": "/opt/sap/FaultManager/dev_sybdbfm",
          "logType": "SAP_ASE_FAULT_MANAGER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
    ],
    "sapAsePrometheusExporter": {
      "sapAseSid": "ASE",
      "sapAsePort": "4901",
      "sapAseSecretName": "ASE_DB_CREDS",
      "prometheusPort": "9399",
      "agreeToEnableASEMonitoring": true
}
```

SAP HANA bei Amazon EC2

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für SAP HANA auf Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "hanadb_server_startup_time_variations_seconds",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_5_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_4_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_out_of_memory_events_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_max_trigger_read_ratio_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_table_allocation_limit_used_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_cpu_usage_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_plan_cache_hit_ratio_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_last_data_backup_age_days",
```

```
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_HANA_TRACE-my-resourge-group",
          "logPath": "/usr/sap/HDB/HDB00/*/trace/*.trc",
          "logType": "SAP_HANA_TRACE",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_HANA_LOGS-my-resource-group",
          "logPath": "/usr/sap/HDB/HDB00/*/trace/*.log",
          "logType": "SAP_HANA_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
      ]
    }
  ],
  "hanaPrometheusExporter": {
    "hanaSid": "HDB",
    "hanaPort": "30013",
    "hanaSecretName": "HANA_DB_CREDS",
    "prometheusPort": "9668"
  }
}
```

SAP HANA bei Amazon EC2 mit hoher Verfügbarkeit

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für SAP HANA auf Amazon EC2 mit hoher Verfügbarkeit.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "hanadb_server_startup_time_variations_seconds",
          "monitor": true
        },
```

```
        {
          "alarmMetricName": "hanadb_level_5_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_4_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_out_of_memory_events_count",
          "monitor": true
        },
        {
          "alarmMetricName": "ha_cluster_pacemaker_stonith_enabled",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_HANA_TRACE-my-resourge-group",
          "logPath": "/usr/sap/HDB/HDB00/*/trace/*.trc",
          "logType": "SAP_HANA_TRACE",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_HANA_HIGH_AVAILABILITY-my-resource-group",
          "logPath": "/var/log/pacemaker/pacemaker.log",
          "logType": "SAP_HANA_HIGH_AVAILABILITY",
          "monitor": true,
          "encoding": "utf-8"
        }
      ]
    }
  ],
  "hanaPrometheusExporter": {
    "hanaSid": "HDB",
    "hanaPort": "30013",
    "hanaSecretName": "HANA_DB_CREDS",
    "prometheusPort": "9668"
  },
  "haClusterPrometheusExporter": {
    "prometheusPort": "9664"
  }
```

```
}
```

SAP NetWeaver auf Amazon EC2

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für SAP NetWeaver auf Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "CPUUtilization",
          "monitor": true
        },
        {
          "alarmMetricName": "StatusCheckFailed",
          "monitor": true
        },
        {
          "alarmMetricName": "disk_used_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "mem_used_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTimeDialog",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTimeDialogRFC",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_DBRequestTime",
          "monitor": true
```

```
        },
        {
          "alarmMetricName": "sap_alerts_LongRunners",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_AbortedJobs",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_BasisSystem",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Database",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Security",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_System",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_QueueTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Availability",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_start_service_processes",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_dispatcher_queue_now",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_dispatcher_queue_max",
          "monitor": true
```

```
        },
        {
          "alarmMetricName": "sap_enqueue_server_locks_max",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_locks_now",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_locks_state",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_replication_state",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_NETWEAVER_DEV_TRACE_LOGS-NetWeaver-ML4",
          "logPath": "/usr/sap/ML4/*/work/dev_w*",
          "logType": "SAP_NETWEAVER_DEV_TRACE_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
      ]
    }
  ],
  "netWeaverPrometheusExporter": {
    "sapSid": "ML4",
    "instanceNumbers": [
      "00",
      "11"
    ],
    "prometheusPort": "9680"
  }
}
```

NetWeaver SAP-Hochverfügbarkeit auf Amazon EC2

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für SAP NetWeaver High Availability auf Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "ha_cluster_corosync_ring_errors",
          "monitor": true
        },
        {
          "alarmMetricName": "ha_cluster_pacemaker_fail_count",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_HA_check_failover_config_state",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_HA_get_failover_config_HAActive",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_AbortedJobs",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Availability",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_BasisSystem",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_DBRequestTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Database",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_FrontendResponseTime",
```

```
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_LongRunners",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_QueueTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTimeDialog",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTimeDialogRFC",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Security",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Shortdumps",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_SqlError",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_System",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_replication_state",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_start_service_processes",
```

```
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_NETWEAVER_DEV_TRACE_LOGS-NetWeaver-PR1",
          "logPath": "/usr/sap/<SID>/D*/work/dev_w*",
          "logType": "SAP_NETWEAVER_DEV_TRACE_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
      ]
    }
  ],
  "haClusterPrometheusExporter": {
    "prometheusPort": "9664"
  },
  "netWeaverPrometheusExporter": {
    "sapSid": "PR1",
    "instanceNumbers": [
      "11",
      "12"
    ],
    "prometheusPort": "9680"
  }
}
```

SQL Always On-Verfügbarkeitsgruppe

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für eine SQL Always On-Verfügbarkeitsgruppe.

```
{
  "subComponents" : [ {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [ {
      "alarmMetricName" : "CPUUtilization",
      "monitor" : true
    }, {
      "alarmMetricName" : "StatusCheckFailed",
      "monitor" : true
    }, {
      "alarmMetricName" : "Processor % Processor Time",
```

```
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory % Committed Bytes In Use",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory Available Mbytes",
      "monitor" : true
    }, {
      "alarmMetricName" : "Paging File % Usage",
      "monitor" : true
    }, {
      "alarmMetricName" : "System Processor Queue Length",
      "monitor" : true
    }, {
      "alarmMetricName" : "Network Interface Bytes Total/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "PhysicalDisk % Disk Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Buffer cache hit ratio",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Page life expectancy",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:General Statistics Processes blocked",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:General Statistics User Connections",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Locks Number of Deadlocks/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:SQL Statistics Batch Requests/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica File Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Bytes Received/sec",
      "monitor" : true
    }, {
```

```
      "alarmMetricName" : "SQLServer:Database Replica Log remaining for undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Send Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Mirrored Write Transaction/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Recovery Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redo Bytes Remaining",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redone Bytes/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Total Log requiring undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Transaction Delay",
      "monitor" : true
    } ],
    "windowsEvents" : [ {
      "logGroupName" : "WINDOWS_EVENTS-Application-<RESOURCE_GROUP_NAME>",
      "eventName" : "Application",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL", "INFORMATION" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-System-<RESOURCE_GROUP_NAME>",
      "eventName" : "System",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-Security-<RESOURCE_GROUP_NAME>",
      "eventName" : "Security",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    } ],
    "logs" : [ {
      "logGroupName" : "SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP-<RESOURCE_GROUP_NAME>",
      "logPath" : "C:\\Program Files\\Microsoft SQL Server\\MSSQL**.MSSQLSERVER\\MSSQL\
\Log\\ERRORLOG",
```

```
      "logType" : "SQL_SERVER",
      "monitor" : true,
      "encoding" : "utf-8"
    } ]
  }, {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [ {
      "alarmMetricName" : "VolumeReadBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeReadOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeQueueLength",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeThroughputPercentage",
      "monitor" : true
    }, {
    "alarmMetricName" : "BurstBalance",
      "monitor" : true
    } ]
  } ]
}
```

SQL-Failoverclusterinstance

Das folgende Beispiel zeigt eine Komponentenkonfiguration im JSON-Format für eine SQL-Failover-Cluster-Instance.

```
{
  "subComponents" : [ {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [ {
      "alarmMetricName" : "CPUUtilization",
      "monitor" : true
    }, {
```

```
      "alarmMetricName" : "StatusCheckFailed",
      "monitor" : true
    }, {
      "alarmMetricName" : "Processor % Processor Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory % Committed Bytes In Use",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory Available Mbytes",
      "monitor" : true
    }, {
      "alarmMetricName" : "Paging File % Usage",
      "monitor" : true
    }, {
      "alarmMetricName" : "System Processor Queue Length",
      "monitor" : true
    }, {
      "alarmMetricName" : "Network Interface Bytes Total/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "PhysicalDisk % Disk Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Normal Messages Queue Length/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Urgent Message Queue Length/se",
      "monitor" : true
    }, {
      "alarmMetricName" : "Reconnect Count",
      "monitor" : true
    }, {
      "alarmMetricName" : "Unacknowledged Message Queue Length/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Messages Outstanding",
      "monitor" : true
    }, {
      "alarmMetricName" : "Messages Sent/sec",
      "monitor" : true
```

```
    }, {
      "alarmMetricName" : "Database Update Messages/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Update Messages/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Flushes/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Crypto Checkpoints Saved/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Crypto Checkpoints Restored/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Registry Checkpoints Restored/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Registry Checkpoints Saved/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Cluster API Calls/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Resource API Calls/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Cluster Handles/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Resource Handles/sec",
      "monitor" : true
    } ],
    "windowsEvents" : [ {
      "logGroupName" : "WINDOWS_EVENTS-Application-<RESOURCE_GROUP_NAME>",
      "eventName" : "Application",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL"],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-System-<RESOURCE_GROUP_NAME>",
      "eventName" : "System",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL", "INFORMATION" ],
      "monitor" : true
```

```
    }, {
      "logGroupName" : "WINDOWS_EVENTS-Security-<RESOURCE_GROUP_NAME>",
      "eventName" : "Security",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    } ],
    "logs" : [ {
      "logGroupName" : "SQL_SERVER_FAILOVER_CLUSTER_INSTANCE-<RESOURCE_GROUP_NAME>",
      "logPath" : "\\\\amznfsxjmzbykwn.mydomain.aws\\SQLDB\\MSSQL**.MSSQLSERVER\\MSSQL\
\Log\\ERRORLOG",
      "logType" : "SQL_SERVER",
      "monitor" : true,
      "encoding" : "utf-8"
    } ]
  }, {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [ {
      "alarmMetricName" : "VolumeReadBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeReadOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeQueueLength",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeThroughputPercentage",
      "monitor" : true
    }, {
    "alarmMetricName" : "BurstBalance",
      "monitor" : true
    } ]
  } ]
}
```

Erstellen und konfigurieren Sie das CloudWatch Application Insights-Monitoring mithilfe von Vorlagen CloudFormation

Sie können Application Insights-Monitoring, einschließlich wichtiger Kennzahlen und Telemetrie, direkt aus AWS CloudFormation Vorlagen zu Ihrer Anwendung, Datenbank und Ihrem Webserver hinzufügen.

Dieser Abschnitt enthält AWS CloudFormation Beispielvorlagen im JSON- und YAML-Format, die Sie bei der Erstellung und Konfiguration der Application Insights-Überwachung unterstützen.

Die Ressourcen- und Eigenschaftsreferenz zu Application Insights im AWS CloudFormation Benutzerhandbuch finden Sie unter Referenz zum ApplicationInsights Ressourcentyp.

Mustervorlagen

- Erstellen Sie eine Application Insights-Anwendung für den gesamten AWS CloudFormation Stack
- Erstellen Sie eine Application-Insights-Anwendung mit detaillierten Einstellungen
- Erstellen Sie eine Application-Insights-Anwendung mit der CUSTOM-Modus-Komponentenkonfiguration
- Erstellen Sie eine Application-Insights-Anwendung mit der DEFAULT-Modus-Komponentenkonfiguration
- Erstellen Sie eine Application-Insights-Anwendung mit der DEFAULT_WITH_OVERWRITE-Modus-Komponentenkonfiguration

Erstellen Sie eine Application Insights-Anwendung für den gesamten AWS CloudFormation Stack

Um die folgende Vorlage anzuwenden, müssen Sie AWS Ressourcen und eine oder mehrere Ressourcengruppen erstellen, aus denen Sie Application Insights-Anwendungen zur Überwachung dieser Ressourcen erstellen können. Weitere Informationen finden Sie unter Erste Schritte mit AWS Resource Groups.

Die ersten beiden Teile der folgenden Vorlage geben eine Ressource und eine Ressourcengruppe an. Im letzten Teil der Vorlage wird eine Application-Insights-Anwendung für die Ressourcengruppe erstellt, die Anwendung jedoch nicht konfiguriert oder die Überwachung angewendet. Weitere Informationen finden Sie in den CreateApplicationBefehlsdetails in der Amazon CloudWatch Application Insights API-Referenz.

Vorlage im JSON-Format

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Test Resource Group stack",
    "Resources": {
        "EC2Instance": {
            "Type": "AWS::EC2::Instance",
            "Properties": {
                "ImageId" : "ami-abcd1234efgh5678i",
                "SecurityGroupIds" : ["sg-abcd1234"]
            }
        },
        ...
        "ResourceGroup": {
            "Type": "AWS::ResourceGroups::Group",
            "Properties": {
                "Name": "my_resource_group"
            }
        },
        "AppInsightsApp": {
            "Type": "AWS::ApplicationInsights::Application",
            "Properties": {
                "ResourceGroupName": "my_resource_group"
            },
            "DependsOn" : "ResourceGroup"
        }
    }
}
```

Vorlage im YAML-Format

```
---
AWSTemplateFormatVersion: '2010-09-09'
Description: Test Resource Group stack
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-abcd1234efgh5678i
      SecurityGroupIds:
      - sg-abcd1234
  ...
```

```
  ResourceGroup:
    Type: AWS::ResourceGroups::Group
    Properties:
      Name: my_resource_group
  AppInsightsApp:
    Type: AWS::ApplicationInsights::Application
    Properties:
      ResourceGroupName: my_resource_group
    DependsOn: ResourceGroup
```

Im folgenden Vorlagenabschnitt wird die Standardüberwachungskonfiguration auf die Application-Insights-Anwendung angewendet. Weitere Informationen finden Sie in den CreateApplicationBefehlsdetails in der Amazon CloudWatch Application Insights API-Referenz.

Wenn `AutoConfigurationEnabled` auf `true` gesetzt ist, werden alle Komponenten der Anwendung mit den empfohlenen Überwachungseinstellungen für die Anwendungsebene `DEFAULT` konfiguriert. Weitere Informationen zu diesen Einstellungen und Stufen finden Sie unter DescribeComponentConfigurationRecommendationund UpdateComponentConfigurationin der Amazon CloudWatch Application Insights API-Referenz.

Vorlage im JSON-Format

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Test Application Insights Application stack",
    "Resources": {
        "AppInsightsApp": {
            "Type": "AWS::ApplicationInsights::Application",
            "Properties": {
                "ResourceGroupName": "my_resource_group",
                "AutoConfigurationEnabled": true
            }
        }
    }
}
```

Vorlage im YAML-Format

```
---
AWSTemplateFormatVersion: '2010-09-09'
Description: Test Application Insights Application stack
Resources:
```

```
  AppInsightsApp:
    Type: AWS::ApplicationInsights::Application
    Properties:
      ResourceGroupName: my_resource_group
      AutoConfigurationEnabled: true
```

Erstellen Sie eine Application-Insights-Anwendung mit detaillierten Einstellungen

Die folgende Vorlage führt diese Aktionen aus:

- Erstellt eine Application Insights-Anwendung mit OpsCenter aktivierter CloudWatch Ereignisbenachrichtigung. Weitere Informationen finden Sie in den CreateApplicationBefehlsdetails in der Amazon CloudWatch Application Insights API-Referenz.
- Markiert die Anwendung mit zwei Tags, von denen eines keine Tag-Werte enthält. Weitere Informationen finden Sie TagResourcein der Amazon CloudWatch Application Insights API-Referenz.
- Erstellt zwei benutzerdefinierte Instance-Gruppenkomponenten. Weitere Informationen finden Sie CreateComponentin der Amazon CloudWatch Application Insights API-Referenz.
- Erstellt zwei Protokollmustersätze. Weitere Informationen finden Sie CreateLogPatternin der Amazon CloudWatch Application Insights API-Referenz.
- Legt `AutoConfigurationEnabled` auf `true` fest, wodurch alle Komponenten der Anwendung mit den empfohlenen Überwachungseinstellungen für die `DEFAULT`-Ebene konfiguriert werden. Weitere Informationen finden Sie DescribeComponentConfigurationRecommendationin der Amazon CloudWatch Application Insights API-Referenz.

Vorlage im JSON-Format

```
{
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
        "ResourceGroupName": "my_resource_group",
        "CWEMonitorEnabled": true,
        "OpsCenterEnabled": true,
        "OpsItemSNSTopicArn": "arn:aws:sns:us-east-1:123456789012:my_topic",
        "AutoConfigurationEnabled": true,
        "Tags": [
            {
                "Key": "key1",
                "Value": "value1"
```

```
            },
            {
                "Key": "key2",
                "Value": ""
            }
        ],
        "CustomComponents": [
            {
                "ComponentName": "test_component_1",
                "ResourceList": [
                    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i"
                ]
            },
            {
                "ComponentName": "test_component_2",
                "ResourceList": [
                    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i",
                    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i"
                ]
            }
        ],
        "LogPatternSets": [
            {
                "PatternSetName": "pattern_set_1",
                "LogPatterns": [
                    {
                        "PatternName": "deadlock_pattern",
                        "Pattern": ".*\\sDeadlocked\\sSchedulers(([^\\w].*)|($))",
                        "Rank": 1
                    }
                ]
            },
            {
                "PatternSetName": "pattern_set_2",
                "LogPatterns": [
                    {
                        "PatternName": "error_pattern",
                        "Pattern": ".*[\\s\\[]ERROR[\\s\\]].*",
                        "Rank": 1
                    },
                    {
                        "PatternName": "warning_pattern",
                        "Pattern": ".*[\\s\\[]WARN(ING)?[\\s\\]].*",
                        "Rank": 10
```

```
                    }
                ]
            }
        ]
    }
}
```

Vorlage im YAML-Format

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
  CWEMonitorEnabled: true
  OpsCenterEnabled: true
  OpsItemSNSTopicArn: arn:aws:sns:us-east-1:123456789012:my_topic
  AutoConfigurationEnabled: true
  Tags:
  - Key: key1
    Value: value1
  - Key: key2
    Value: ''
  CustomComponents:
  - ComponentName: test_component_1
    ResourceList:
    - arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i
  - ComponentName: test_component_2
    ResourceList:
    - arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i
    - arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i
  LogPatternSets:
  - PatternSetName: pattern_set_1
    LogPatterns:
    - PatternName: deadlock_pattern
      Pattern: ".*\\sDeadlocked\\sSchedulers(([^\\w].*)|($))"
      Rank: 1
  - PatternSetName: pattern_set_2
    LogPatterns:
    - PatternName: error_pattern
      Pattern: ".*[\\s\\[]ERROR[\\s\\]].*"
      Rank: 1
    - PatternName: warning_pattern
      Pattern: ".*[\\s\\[]WARN(ING)?[\\s\\]].*"
```

```
      Rank: 10
```

Erstellen Sie eine Application-Insights-Anwendung mit der **CUSTOM**-Modus-Komponentenkonfiguration

Die folgende Vorlage führt diese Aktionen aus:

- Erstellt eine Application Insights-Anwendung. Weitere Informationen finden Sie CreateApplicationin der Amazon CloudWatch Application Insights API-Referenz.
- Mit Komponente `my_component` wird `ComponentConfigurationMode` auf `CUSTOM` festgelegt, wodurch diese Komponente wie in `CustomComponentConfiguration` angegeben mit der Konfiguration konfiguriert wird. Weitere Informationen finden Sie UpdateComponentConfigurationin der Amazon CloudWatch Application Insights API-Referenz.

Vorlage im JSON-Format

```
{
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
        "ResourceGroupName": "my_resource_group,
        "ComponentMonitoringSettings": [
            {
                "ComponentARN": "my_component",
                "Tier": "SQL_SERVER",
                "ComponentConfigurationMode": "CUSTOM",
                "CustomComponentConfiguration": {
                    "ConfigurationDetails": {
                        "AlarmMetrics": [
                            {
                                "AlarmMetricName": "StatusCheckFailed"
                            },
                            ...
                        ],
                        "Logs": [
                            {
                                "LogGroupName": "my_log_group_1",
                                "LogPath": "C:\\LogFolder_1\\*",
                                "LogType": "DOT_NET_CORE",
                                "Encoding": "utf-8",
                                "PatternSet": "my_pattern_set_1"
                            },
```

```
                        ...
                    ],
                    "WindowsEvents": [
                        {
                            "LogGroupName": "my_windows_event_log_group_1",
                            "EventName": "Application",
                            "EventLevels": [
                                "ERROR",
                                "WARNING",
                                ...
                            ],
                            "Encoding": "utf-8",
                            "PatternSet": "my_pattern_set_2"
                        },
                        ...
                    ],
                    "Alarms": [
                        {
                            "AlarmName": "my_alarm_name",
                            "Severity": "HIGH"
                        },
                        ...
                    ]
                },
                "SubComponentTypeConfigurations": [
                    {
                        "SubComponentType": "EC2_INSTANCE",
                        "SubComponentConfigurationDetails": {
                            "AlarmMetrics": [
                                {
                                    "AlarmMetricName": "DiskReadOps"
                                },
                                ...
                            ],
                            "Logs": [
                                {
                                    "LogGroupName": "my_log_group_2",
                                    "LogPath": "C:\\LogFolder_2\\*",
                                    "LogType": "IIS",
                                    "Encoding": "utf-8",
                                    "PatternSet": "my_pattern_set_3"
                                },
                                ...
                            ],
```

```
                                "processes" : [
                                    {
                                        "processName" : "my_process",
                                        "alarmMetrics" : [
                                    {
                                        "alarmMetricName" : "procstat cpu_usage",
                                        "monitor" : true
                                    }, {
                                        "alarmMetricName" : "procstat memory_rss",
                                        "monitor" : true
                                    }
                                ]
                            }
                        ],
                                "WindowsEvents": [
                                    {
                                        "LogGroupName": "my_windows_event_log_group_2",
                                        "EventName": "Application",
                                        "EventLevels": [
                                            "ERROR",
                                            "WARNING",
                                            ...
                                        ],
                                        "Encoding": "utf-8",
                                        "PatternSet": "my_pattern_set_4"
                                    },
                                    ...
                                ]
                            }
                        }
                    ]
                }
            }
        ]
    }
}
```

Vorlage im YAML-Format

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
```

```
  ComponentMonitoringSettings:
  - ComponentARN: my_component
    Tier: SQL_SERVER
    ComponentConfigurationMode: CUSTOM
    CustomComponentConfiguration:
      ConfigurationDetails:
        AlarmMetrics:
        - AlarmMetricName: StatusCheckFailed
        ...
        Logs:
        - LogGroupName: my_log_group_1
          LogPath: C:\LogFolder_1\*
          LogType: DOT_NET_CORE
          Encoding: utf-8
          PatternSet: my_pattern_set_1
        ...
        WindowsEvents:
        - LogGroupName: my_windows_event_log_group_1
          EventName: Application
          EventLevels:
          - ERROR
          - WARNING
          ...
          Encoding: utf-8
          PatternSet: my_pattern_set_2
        ...
        Alarms:
        - AlarmName: my_alarm_name
          Severity: HIGH
        ...
      SubComponentTypeConfigurations:
      - SubComponentType: EC2_INSTANCE
        SubComponentConfigurationDetails:
          AlarmMetrics:
          - AlarmMetricName: DiskReadOps
          ...
          Logs:
          - LogGroupName: my_log_group_2
            LogPath: C:\LogFolder_2\*
            LogType: IIS
            Encoding: utf-8
            PatternSet: my_pattern_set_3
          ...
          Processes:
```

```
          - ProcessName: my_process
            AlarmMetrics:
            - AlarmMetricName: procstat cpu_usage
              ...
            ...
          WindowsEvents:
          - LogGroupName: my_windows_event_log_group_2
            EventName: Application
            EventLevels:
            - ERROR
            - WARNING
            ...
            Encoding: utf-8
            PatternSet: my_pattern_set_4
          ...
```

Erstellen Sie eine Application-Insights-Anwendung mit der **DEFAULT**-Modus-Komponentenkonfiguration

Die folgende Vorlage führt diese Aktionen aus:

- Erstellt eine Application Insights-Anwendung. Weitere Informationen finden Sie CreateApplicationin der Amazon CloudWatch Application Insights API-Referenz.
- Mit Komponente `my_component` wird `ComponentConfigurationMode` auf `DEFAULT` und `Tier` auf `SQL_SERVER` festgelegt, wodurch diese Komponente mit den Konfigurationseinstellungen konfiguriert wird, die Application Insights für Ebene `SQL_Server` empfiehlt. Weitere Informationen finden Sie unter DescribeComponentConfigurationund UpdateComponentConfigurationin der Amazon CloudWatch Application Insights API-Referenz.

Vorlage im JSON-Format

```
{
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
        "ResourceGroupName": "my_resource_group",
        "ComponentMonitoringSettings": [
            {
                "ComponentARN": "my_component",
                "Tier": "SQL_SERVER",
                "ComponentConfigurationMode": "DEFAULT"
            }
```

```
        ]
    }
}
```

Vorlage im YAML-Format

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
  ComponentMonitoringSettings:
  - ComponentARN: my_component
    Tier: SQL_SERVER
    ComponentConfigurationMode: DEFAULT
```

Erstellen Sie eine Application-Insights-Anwendung mit der **`DEFAULT_WITH_OVERWRITE`**-Modus-Komponentenkonfiguration

Die folgende Vorlage führt diese Aktionen aus:

- Erstellt eine Application Insights-Anwendung. Weitere Informationen finden Sie CreateApplicationin der Amazon CloudWatch Application Insights API-Referenz.
- Mit Komponente `my_component` wird `ComponentConfigurationMode` auf `DEFAULT_WITH_OVERWRITE` und `tier` auf `DOT_NET_CORE` festgelegt, wodurch diese Komponente mit den Konfigurationseinstellungen konfiguriert wird, die Application Insights für Ebene `DOT_NET_CORE` empfiehlt. Außer Kraft gesetzte Konfigurationseinstellungen werden in `DefaultOverwriteComponentConfiguration` angegeben:
  - Auf Komponentenebene werden `AlarmMetrics`-Einstellungen außer Kraft gesetzt.
  - Auf der Unterkomponentenebene werden die `Logs`-Einstellungen für die Unterkomponenten des `EC2_Instance`-Typs außer Kraft gesetzt.

  Weitere Informationen finden Sie UpdateComponentConfigurationin der Amazon CloudWatch Application Insights API-Referenz.

Vorlage im JSON-Format

```
{
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
```

```
        "ResourceGroupName": "my_resource_group",
        "ComponentMonitoringSettings": [
            {
                "ComponentName": "my_component",
                "Tier": "DOT_NET_CORE",
                "ComponentConfigurationMode": "DEFAULT_WITH_OVERWRITE",
                "DefaultOverwriteComponentConfiguration": {
                    "ConfigurationDetails": {
                        "AlarmMetrics": [
                            {
                                "AlarmMetricName": "StatusCheckFailed"
                            }
                        ]
                    },
                    "SubComponentTypeConfigurations": [
                        {
                            "SubComponentType": "EC2_INSTANCE",
                            "SubComponentConfigurationDetails": {
                                "Logs": [
                                    {
                                        "LogGroupName": "my_log_group",
                                        "LogPath": "C:\\LogFolder\\*",
                                        "LogType": "IIS",
                                        "Encoding": "utf-8",
                                        "PatternSet": "my_pattern_set"
                                    }
                                ]
                            }
                        }
                    ]
                }
            }
        ]
    }
}
```

Vorlage im YAML-Format

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
  ComponentMonitoringSettings:
```

```
  - ComponentName: my_component
    Tier: DOT_NET_CORE
    ComponentConfigurationMode: DEFAULT_WITH_OVERWRITE
    DefaultOverwriteComponentConfiguration:
      ConfigurationDetails:
        AlarmMetrics:
        - AlarmMetricName: StatusCheckFailed
      SubComponentTypeConfigurations:
      - SubComponentType: EC2_INSTANCE
        SubComponentConfigurationDetails:
          Logs:
          - LogGroupName: my_log_group
            LogPath: C:\LogFolder\*
            LogType: IIS
            Encoding: utf-8
            PatternSet: my_pattern_set
```

Tutorial: Einrichten der Überwachung für SAP ASE

Dieses Tutorial zeigt, wie Sie CloudWatch Application Insights konfigurieren, um die Überwachung Ihrer SAP ASE-Datenbanken einzurichten. Sie können die automatischen Dashboards von CloudWatch Application Insights verwenden, um Problemdetails zu visualisieren, die Fehlerbehebung zu beschleunigen und die Mean Time to Resolution (MTTR) für Ihre SAP ASE-Datenbanken zu vereinfachen.

Application Insights für SAP-ASE-Themen

- Unterstützte Umgebungen
- Unterstützte Betriebssysteme
- Features
- Voraussetzungen
- Überwachung Ihrer SAP-ASE-Datenbank einrichten
- Verwalten der Überwachung der SAP-ASE-Datenbank
- Konfigurieren des Alarmschwellenwerts
- Von Application Insights erkannte Probleme mit SAP ASE anzeigen und beheben
- Fehlerbehebung bei Application Insights für SAP ASE

Unterstützte Umgebungen

CloudWatch Application Insights unterstützt die Bereitstellung von AWS Ressourcen für die folgenden Systeme und Muster. Sie stellen die SAP-ASE-Datenbanksoftware und unterstützte SAP-Anwendungssoftware bereit und installieren sie.

- Eine oder mehrere SAP-ASE-Datenbanken auf einer einzelnen Amazon-EC2-Instance – SAP ASE in einer Architektur zum Hochskalieren mit einem einzigen Knoten.
- Cross-AZ-SAP-ASE-Datenbankeinrichtung mit hoher Verfügbarkeit – SAP ASE mit hoher Verfügbarkeit, konfiguriert über zwei Availability Zones unter Verwendung von SUSE/RHEL-Clustering.

> **Note**
>
> CloudWatch Application Insights unterstützt nur einzelne SAP System ID (SID) ASE HA-Umgebungen. Wenn mehrere ASE-HA-SIDs angeschlossen sind, wird die Überwachung nur für die erste erkannte SID eingerichtet.

Unterstützte Betriebssysteme

CloudWatch Application Insights for SAP ASE unterstützt die x86-64-Architektur auf den folgenden Betriebssystemen:

- SuSE Linux 12 SP4
- SuSE Linux 12 SP5
- SuSE Linux 15
- SuSE Linux 15 SP1
- SuSE Linux 15 SP2
- SuSE Linux 15 SP3
- SuSE Linux 15 SP4
- SuSE Linux 15 SP1 für SAP
- SuSE Linux 15 SP2 für SAP
- SuSE Linux 15 SP3 für SAP
- SuSE Linux 15 SP4 für SAP

- SuSE Linux 12 SP4 für SAP
- SuSE Linux 12 SP5 für SAP
- RedHat Linux 7.6
- RedHat Linux 7.7
- RedHat Linux 7.9
- RedHat Linux 8.1
- RedHat Linux 8.4
- RedHat Linux 8.6

Features

CloudWatch Application Insights for SAP ASE bietet die folgenden Funktionen:

- Automatische SAP-ASE-Workload-Erkennung
- Automatische SAP-ASE-Alarmerstellung basierend auf statischem Schwellenwert
- Automatische SAP-ASE-Alarmerstellung basierend auf Anomalieerkennung
- Automatische SAP-ASE-Protokoll-Mustererkennung
- Zustands-Dashboard für SAP ASE
- Problem-Dashboard für SAP ASE

Voraussetzungen

Sie müssen die folgenden Voraussetzungen erfüllen, um eine SAP ASE-Datenbank mit CloudWatch Application Insights zu konfigurieren:

- SAP-ASE-Konfigurationsparameter – Die folgenden Konfigurationsparameter müssen in Ihrer ASE-DB aktiviert sein: `"enable monitoring"`, `"sql text pipe max messages"`, `"sql text pipe active"`. Auf diese Weise kann CloudWatch Application Insights umfassende Überwachungsfunktionen für Ihre Datenbank bereitstellen. Wenn diese Einstellungen in Ihrer ASE-Datenbank nicht aktiviert sind, werden sie von Application Insights automatisch aktiviert, um die notwendigen Metriken für die Überwachung zu sammeln.
- SAP-ASE-Datenbankbenutzer – Der Datenbankbenutzer, den Sie beim Onboarding von Application Insights angegeben haben, muss über Zugriffsberechtigungen für Folgendes verfügen:
  - Systemtabellen in der Master-Datenbank und in den Benutzer-(Tenant-)Datenbanken

  - Überwachungstabellen
- SAP HostCtrl — Installieren und richten Sie SAP HostCtrl auf Ihrer Amazon EC2 EC2-Instance ein.
- CloudWatch Amazon-Agent — Stellen Sie sicher, dass Sie keinen bereits vorhandenen CloudWatch Agenten auf Ihrer Amazon EC2-Instance ausführen. Wenn Sie einen CloudWatch Agenten installiert haben, stellen Sie sicher, dass Sie die Konfiguration der Ressourcen, die Sie in CloudWatch Application Insights verwenden, aus der vorhandenen CloudWatch Agenten-Konfigurationsdatei entfernen, um einen Zusammenführungskonflikt zu vermeiden. Weitere Informationen finden Sie unter Erstellen oder bearbeiten Sie die CloudWatch Agenten-Konfigurationsdatei manuell.
- AWS Systems Manager Manager-Aktivierung — Installieren Sie den SSM Agent auf Ihren Instances und aktivieren Sie die für SSM aktivierten Instanzen. Informationen über das Installieren des SSM Agent finden Sie unter Arbeiten mit dem SSM Agent im AWS -Systems-Manager-Benutzerhandbuch.
- Rollen für Amazon-EC2-Instances – Sie müssen die folgenden Amazon-EC2-Instance-Rollen anhängen, um Ihre Datenbank zu konfigurieren.
  - Sie müssen die `AmazonSSMManagedInstanceCore`-Rolle anfügen, um Systems Manager zu aktivieren. Weitere Informationen finden Sie unter Beispiele für identitätsbasierte AWS Systems Manager -Richtlinien.
  - Sie müssen das anhängen`CloudWatchAgentServerPolicy`, damit Instanzmetriken und -protokolle ausgegeben werden können. CloudWatch Weitere Informationen finden Sie unter IAM-Rollen und -Benutzer für die Verwendung mit Amazon CloudWatch Agent erstellen.
  - Sie müssen die folgende IAM-Inline-Richtlinie an die Amazon-EC2-Instancerolle anhängen, um das in AWS Secrets Manager gespeicherte Passwort auszulesen. Weitere Informationen zu Inline-Richtlinien finden Sie unter Inline-Richtlinien im AWS Identity and Access Management -Benutzerhandbuch.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
        }
```

```
    ]
}
```

- AWS Resource Groups— Sie müssen eine Ressourcengruppe erstellen, die alle zugehörigen AWS Ressourcen umfasst, die von Ihrem Anwendungsstapel verwendet werden, um Ihre Anwendungen in CloudWatch Application Insights zu integrieren. Dies umfasst Amazon-EC2-Instances und Amazon-EBS-Volumes, auf denen Ihre SAP-ASE-Datenbank ausgeführt wird. Wenn es mehrere Datenbanken pro Konto gibt, empfehlen wir, dass Sie eine Ressourcengruppe erstellen, die die AWS Ressourcen für jedes SAP ASE-Datenbanksystem enthält.
- IAM-Berechtigungen – Für Nicht-Admin-Benutzer:
  - Sie müssen eine AWS Identity and Access Management (IAM-) Richtlinie erstellen, die es Application Insights ermöglicht, eine dienstbezogene Rolle zu erstellen, und sie Ihrer Benutzeridentität zuordnen. Wie Sie die Richtlinie anfügen, erfahren Sie unter IAM-Richtlinie.
  - Der Benutzer muss berechtigt sein, ein Geheimnis zu erstellen, in AWS Secrets Manager dem die Anmeldeinformationen des Datenbankbenutzers gespeichert werden. Weitere Informationen finden Sie unter Beispiel: Berechtigung zum Erstellen von Secrets.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:CreateSecret"
      ],
      "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
    }
  ]
}
```

- Dienstverknüpfte Rolle — Application Insights verwendet AWS Identity and Access Management (IAM) dienstbezogene Rollen. Es wird für Sie eine serviceverknüpfte Rolle erstellt, wenn Sie Ihre erste Application-Insights-Anwendung in der Application-Insights-Konsole erstellen. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für CloudWatch Application Insights.

Überwachung Ihrer SAP-ASE-Datenbank einrichten

Gehen Sie wie folgt vor, um die Überwachung für Ihre SAP-ASE-Datenbank einzurichten

1. Öffnen Sie die CloudWatch-Konsole.
2. Wählen Sie im linken Navigationsbereich unter Insights die Option Application Insights aus.
3. Auf der Seite Application Insights sehen Sie die Anwendungen, die mit Application Insights überwacht werden, samt Überwachungsstatus. Wählen Sie in der oberen rechten Ecke Add an application (Eine Anwendung hinzufügen) aus.
4. Auf der Seite Anwendungsdetails angeben wählen Sie aus der Dropdown-Liste unter Ressourcengruppe die AWS -Ressourcengruppe aus, die Ihre SAP-ASE-Datenbankressourcen enthält. Wenn Sie keine Ressourcengruppe für Ihre Anwendung erstellt haben, können Sie das mit Create new resource group (Neue Ressourcengruppe erstellen) im Dropdown-Menü Resource group (Ressourcengruppe) tun. Weitere Informationen zu Ressourcengruppen finden Sie im AWS -Benutzerhandbuch zu Resource Groups.
5. Aktivieren Sie unter CloudWatch Ereignisse überwachen das Kontrollkästchen, um die Überwachung von Application Insights mit CloudWatch Ereignissen zu integrieren, um Erkenntnisse aus Amazon EBS, Amazon EC2, Amazon ECS AWS CodeDeploy, AWS Health APIs und Benachrichtigungen, Amazon RDS, Amazon S3 und zu erhalten. AWS Step Functions
6. Aktivieren Sie unter Integrieren mit AWS Systems Manager OpsCenter das Kontrollkästchen neben AWS Systems Manager OpsCenter OpsItems Für Abhilfemaßnahmen generieren, um Benachrichtigungen anzuzeigen und zu erhalten, wenn bei den ausgewählten Anwendungen Probleme festgestellt werden. Geben Sie einen ARN für das SNS-Thema an, um die Operationen nachzuverfolgen OpsItems, die zur Lösung von so genannten operativen Arbeitsaufgaben ausgeführt werden, die sich auf Ihre AWS Ressourcen beziehen.
7. Sie können optional Tags eingeben, die Ihnen helfen, Ihre Ressourcen zu identifizieren und zu organisieren. CloudWatch Application Insights unterstützt sowohl tagbasierte als auch AWS CloudFormation stapelbasierte Ressourcengruppen, mit Ausnahme von Gruppen. Application Auto Scaling Weitere Informationen finden Sie unter Tag-Editor im Benutzerhandbuch zu AWS Resource Groups und Tags.
8. Wählen Sie Next (Weiter) aus, um mit der Einrichtung der Überwachung fortzufahren.
9. Auf der Seite Entdeckte Komponenten überprüfen werden die überwachten Komponenten und ihre Workloads aufgeführt, die von CloudWatch Application Insights automatisch erkannt wurden.

> **Note**
>
> Komponenten, die einen erkannten Workload von SAP ASE High Availability enthalten, unterstützen nur einen Workload auf einer Komponente. Komponenten, die einen erkannten SAP-ASE-Workload mit einem einzelnen Knoten enthalten, unterstützen

mehrere Workloads, aber Sie können keine Workloads hinzufügen oder entfernen. Alle automatisch erkannten Workloads werden überwacht.

10. Wählen Sie Weiter aus.
11. Geben Sie auf der Seite Komponentendetails angeben den Benutzernamen und das Passwort Ihrer SAP-ASE-Datenbanken ein.
12. Überprüfen Sie die Konfiguration der Anwendungsüberwachung und wählen Sie Submit (Absenden) aus.
13. Es öffnet sich die Seite mit den Anwendungsdetails, auf der Sie die Anwendungsübersicht, die Liste der überwachten Komponenten und Workloads sowie der nicht überwachten Komponenten und Workloads sehen können. Wenn Sie das Optionsfeld neben einer Komponente oder einem Workload auswählen, können Sie auch den Konfigurationsverlauf, die Protokollmuster und alle von Ihnen erstellten Tags anzeigen. Wenn Sie Ihre Konfiguration einreichen, stellt Ihr Konto alle Metriken und Alarme für Ihr SAP-ASE-System bereit, was bis zu 2 Stunden dauern kann.

Verwalten der Überwachung der SAP-ASE-Datenbank

Sie können Benutzeranmeldeinformationen, Metriken und Protokollpfade für Ihre SAP-ASE-Datenbank verwalten, indem Sie die folgenden Schritte ausführen:

1. Öffnen Sie die CloudWatch-Konsole.
2. Wählen Sie im linken Navigationsbereich unter Insights die Option Application Insights aus.
3. Auf der Seite Application Insights sehen Sie die Anwendungen, die mit Application Insights überwacht werden, samt Überwachungsstatus.
4. Unter Monitored components (Überwachte Komponenten) wählen Sie das Optionsfeld neben dem Komponentennamen aus. Wählen Sie dann Manage monitoring (Überwachung verwalten) aus.
5. Unter EC2 instance group logs (Protokolle für EC2-Instance-Gruppen) können Sie den vorhandenen Protokollpfad, den Protokollmustersatz und den Namen der Protokollgruppe bearbeiten. Darüber hinaus können Sie bis zu drei zusätzliche Application logs (Anwendungsprotokolle) hinzufügen.
6. Unter Metriken können Sie die SAP-ASE-Metriken nach Ihren Anforderungen auswählen. SAP-ASE-Metriknamen haben das Präfix `asedb`. Sie können bis zu 60 Metriken pro Komponente hinzufügen.

7. Geben Sie unter ASE-Konfiguration das Passwort und den Benutzernamen für die SAP-ASE-Datenbank ein. Dies sind der Benutzername und das Passwort, die der CloudWatch Amazon-Agent verwendet, um sich mit der SAP ASE-Datenbank zu verbinden.
8. Unter Benutzerdefinierte Alarme können Sie zusätzliche Alarme hinzufügen, die von CloudWatch Application Insights überwacht werden sollen.
9. Überprüfen Sie die Konfiguration der Anwendungsüberwachung und wählen Sie Submit (Absenden) aus. Wenn Sie Ihre Konfiguration einreichen, aktualisiert Ihr Konto alle Metriken und Alarme für Ihr SAP HANA-System, was bis zu 2 Stunden dauern kann.

Konfigurieren des Alarmschwellenwerts

CloudWatch Application Insights erstellt automatisch eine CloudWatch Amazon-Metrik, die der Alarm überwacht, zusammen mit dem Schwellenwert für diese Metrik. Der Alarm wechselt in den Status `ALARM` , wenn die Metrik für eine bestimmte Anzahl von Auswertungszeiträumen den Schwellenwert überschreitet. Beachten Sie, dass diese Einstellungen von Application Insights nicht beibehalten werden.

Um einen Alarm für eine einzelne Metrik zu bearbeiten, tun Sie Folgendes:

1. Öffnen Sie die CloudWatch-Konsole.
2. Wählen Sie im Navigationsbereich Alarms (Alarme) und All alarms (Alle Alarme) aus.
3. Wählen Sie das Optionsfeld neben dem Alarm aus, der automatisch von CloudWatch Application Insights erstellt wurde. Wählen Sie dann Actions (Aktionen) aus und dann Edit (Bearbeiten) im Dropdown-Menü.
4. Bearbeiten Sie die folgenden Parameter unter Metric (Metrik).
    a. Wählen Sie unter Statistic (Statistik) eine der Statistiken oder vordefinierten Perzentile aus, oder geben Sie ein benutzerdefiniertes Perzentil an. z. B. `p95.45`.
    b. Wählen Sie unter Period (Zeitraum) den Auswertungszeitraum für den Alarm aus. Beim Auswerten des Alarms wird jeder Zeitraum in einem Datenpunkt zusammengefasst.
5. Bearbeiten Sie die folgenden Parameter unter Conditions (Bedingungen).
    a. Geben Sie an, ob die Metrik größer, kleiner oder gleich dem Schwellenwert sein muss.
    b. Geben Sie den Schwellenwert an.
6. Bearbeiten Sie unter Additional configuration (Zusätzliche Konfiguration) die folgenden Parameter.

a. Geben Sie unter Datapoints to alarm (Zu alarmierende Datenpunkte) die Anzahl der Datenpunkte bzw. Auswertungszeiträume an, die im Status `ALARM` sein müssen, um den Alarm auszulösen. Wenn die beiden Werte übereinstimmen, wird ein Alarm erstellt, der in den Status `ALARM` wechselt, wenn die festgelegte Anzahl aufeinander folgender Zeiträume überschritten wird. Um einen `m`-von-n-Alarm zu erstellen, geben Sie für den ersten Datenpunkt eine niedrigere Zahl als für den zweiten Datenpunkt an. Weitere Informationen zum Auswerten von Alarmen finden Sie unter Auswerten eines Alarms.

b. Wählen Sie unter Missing data treatment (Behandlung von fehlenden Daten) aus, wie sich der Alarm verhalten soll, wenn einige Datenpunkte fehlen. Weitere Informationen zur Behandlung fehlender Daten finden Sie unter Konfiguration der Behandlung fehlender Daten durch CloudWatch Alarme.

c. Wenn der Alarm ein Perzentil als überwachte Statistik verwendet, erscheint ein Feld Percentiles with low samples (Perzentile mit geringen Stichproben). Entscheiden Sie, ob Sie Fälle mit niedrigem Stichprobenumfang bewerten oder ignorieren möchten. Wenn Sie ignore (maintain alarm state) (Ignorieren (Alarmstatus beibehalten)) wählen, wird der aktuelle Alarmstatus immer beibehalten, wenn die Stichprobengröße zu gering ist. Weitere Informationen zu Perzentilen mit wenigen Beispielen finden Sie unter Auf Perzentilen basierende CloudWatch Alarme und Stichproben mit niedrigen Datenmengen.

7. Wählen Sie Next (Weiter) aus.
8. Wählen Sie unter Notification (Benachrichtigung) ein SNS-Thema aus, das benachrichtigt werden soll, wenn sich der Alarm im Status `ALARM`, `OK` oder `INSUFFICIENT_DATA` befindet.
9. Wählen Sie Update Alarm (Alarm bearbeiten) aus.

Von Application Insights erkannte Probleme mit SAP ASE anzeigen und beheben

Dieser Abschnitt hilft Ihnen bei der Behebung gängiger Probleme, die bei der Konfiguration der Überwachung für SAP ASE auf Application Insights auftreten.

SAP-ASE-Backup-Serverfehler

Sie können die Fehlermeldung identifizieren, indem Sie das dynamisch erstellte Dashboard überprüfen. Das Dashboard zeigt die im SAP-ASE-Backup-Server gemeldete Fehlermeldung an. Weitere Informationen zu Protokollen für SAP-ASE-Backup-Server finden Sie unter SAP-Dokumentation Backup-Server-Fehlerprotokollierung.

SAP-ASE-Transaktionen mit langer Laufzeit

Identifizieren Sie die Transaktion mit langer Laufzeit und überprüfen Sie, ob sie gestoppt werden kann oder ob die Laufzeit beabsichtigt ist. Weitere Informationen finden Sie unter 2180410 – Wie werden Transaktionsprotokolleinträge für Transaktionen mit langer Laufzeit angezeigt? – SAP ASE.

SAP-ASE-Benutzerverbindungen

Prüfen Sie, ob Ihre SAP-ASE-Datenbank entsprechend des Workloads dimensioniert ist, die Sie auf der Datenbank ausführen möchten. Weitere Informationen finden Sie in der SAP-Dokumentation unter Benutzerverbindungen konfigurieren.

SAP-ASE-Festplattenspeicher

Sie können die Datenbankebene identifizieren, die das Problem verursacht, indem Sie das dynamisch erstellte Dashboard überprüfen. Das Dashboard zeigt die entsprechenden Metriken und Auszüge aus den Protokolldateien an. Es ist wichtig, die Ursache für das Festplattenwachstum zu verstehen und gegebenenfalls die physische Festplattengröße, den zugewiesenen Festplattenspeicher oder beides zu erhöhen. Weitere Informationen finden Sie unter SAP-Dokumentation zur Änderung der Festplattengröße in der SAP-Dokumentation.

Fehlerbehebung bei Application Insights für SAP ASE

Dieser Abschnitt enthält Schritte, mit denen Sie häufige Fehler beheben können, die vom Application-Insights-Dashboard zurückgegeben werden.

Fehler	Zurückgegebener Fehler	Ursache	Auflösung
Es können nicht mehr als 60 Überwachungsmetriken hinzugefügt werden.	`Component cannot have more than 60 monitored metric`	Das aktuelle Metriklimit liegt bei 60 überwachten Metriken pro Komponente.	Entfernen Sie unnötige Metriken, um das Limit einzuhalten.
Nach dem Onboarding-Prozess werden keine SAP-Metriken oder Alarme angezeigt	Der Befehl `run` auf `AWS-ConfigureAWSPackage` ist in AWS Systems Manager fehlgeschlagen.	Der Benutzername und das Passwort sind möglicherweise falsch.	Vergewissern Sie sich, dass der Benutzername und das Passwort gültig sind, und führen Sie dann den Onboardin

Fehler	Zurückgegebener Fehler	Ursache	Auflösung
	Die Ausgabe zeigt folgenden Fehler: `CT-LIBRARY error:ct_connect(): protocol specific layer: external error: The attempt to connect to the server failed`		g-Prozess erneut durch.

Praktische Anleitung: Einrichten der Überwachung für SAP HANA

Dieses Tutorial zeigt, wie Sie CloudWatch Application Insights konfigurieren, um die Überwachung Ihrer SAP HANA-Datenbanken einzurichten. Sie können die automatischen Dashboards von CloudWatch Application Insights verwenden, um Problemdetails zu visualisieren, die Fehlerbehebung zu beschleunigen und die Mean Time to Resolution (MTTR) für Ihre SAP HANA-Datenbanken zu vereinfachen.

Application Insights für SAP HANA-Themen

- Unterstützte Umgebungen
- Unterstützte Betriebssysteme
- Features
- Voraussetzungen
- Einrichten der SAP HANA-Datenbank für die Überwachung
- Verwalten der Überwachung der SAP HANA-Datenbank
- Von CloudWatch Application Insights erkannte SAP HANA-Probleme anzeigen und beheben
- Erkennung von Anomalien für SAP HANA
- Fehlerbehebung bei Application Insights für SAP HANA

Unterstützte Umgebungen

CloudWatch Application Insights unterstützt die Bereitstellung von AWS Ressourcen für die folgenden Systeme und Muster. Sie stellen die SAP HANA-Datenbanksoftware und unterstützte SAP-Anwendungssoftware bereit und installieren sie.

- SAP HANA-Datenbank auf einer einzelnen Amazon-EC2-Instance – SAP HANA in einer Scale-Up-Architektur mit einem einzigen Knoten mit bis zu 24 TB Speicher.
- SAP HANA-Datenbank auf mehreren Amazon-EC2-Instances – SAP HANA in einer Scale-Out-Architektur mit mehreren Knoten.
- Cross-AZ-SAP HANA-Datenbankeinrichtung, hohe Verfügbarkeit – SAP HANA mit hoher Verfügbarkeit, konfiguriert über zwei Availability Zones unter Verwendung von SUSE/RHEL-Clustering.

> **Note**
>
> CloudWatch Application Insights unterstützt nur einzelne SID-HANA-Umgebungen. Wenn mehrere HANA-SIDs angeschlossen sind, wird die Überwachung nur für die erste erkannte SID eingerichtet.

Unterstützte Betriebssysteme

CloudWatch Application Insights for SAP HANA unterstützt die x86-64-Architektur auf den folgenden Betriebssystemen:

- SuSE Linux 12 SP4 für SAP
- SuSE Linux 12 SP5 für SAP
- SuSE Linux 15
- SuSE Linux 15 SP1
- SuSE Linux 15 SP2
- SuSE Linux 15 für SAP
- SuSE Linux 15 SP1 für SAP
- SuSE Linux 15 SP2 für SAP
- SuSE Linux 15 SP3 für SAP

- SuSE Linux 15 SP4 für SAP
- SuSE Linux 15 SP5 für SAP
- RedHat Linux 8.6 für SAP mit Hochverfügbarkeit und Aktualisierungsdiensten
- RedHat Linux 8.5 für SAP mit Hochverfügbarkeit und Aktualisierungsdiensten
- RedHat Linux 8.4 für SAP mit Hochverfügbarkeit und Aktualisierungsdiensten
- RedHat Linux 8.3 für SAP mit Hochverfügbarkeit und Aktualisierungsdiensten
- RedHat Linux 8.2 für SAP mit Hochverfügbarkeit und Aktualisierungsdiensten
- RedHat Linux 8.1 für SAP mit Hochverfügbarkeit und Aktualisierungsdiensten
- RedHat Linux 7.9 für SAP mit Hochverfügbarkeit und Aktualisierungsdiensten

Features

CloudWatch Application Insights für SAP HANA bietet die folgenden Funktionen:

- Automatische SAP HANA-Workload-Erkennung
- Automatische SAP HANA-Alarmerstellung basierend auf statischem Schwellenwert
- Automatische SAP HANA-Alarmerstellung basierend auf Anomalieerkennung
- Automatische SAP HANA Log-Mustererkennung
- Zustands-Dashboard für SAP HANA
- Problem-Dashboard für SAP HANA

Voraussetzungen

Sie müssen die folgenden Voraussetzungen erfüllen, um eine SAP HANA-Datenbank mit CloudWatch Application Insights zu konfigurieren:

- SAP HANA — Installieren Sie eine laufende und erreichbare SAP HANA-Datenbank 2.0 SPS05 auf einer Amazon EC2 EC2-Instance.
- SAP HANA-Datenbankbenutzer — Ein Datenbankbenutzer mit Überwachungsrollen muss in der SYSTEM-Datenbank und allen Mandanten erstellt werden.

  Beispiel

  Mit den folgenden SQL-Befehlen erstellen Sie einen Benutzer mit Überwachungsrollen.

```
su - <sid>adm
hdbsql -u SYSTEM -p <SYSTEMDB password> -d SYSTEMDB
CREATE USER CW_HANADB_EXPORTER_USER PASSWORD <Monitoring user password> NO
 FORCE_FIRST_PASSWORD_CHANGE;
CREATE ROLE CW_HANADB_EXPORTER_ROLE;
GRANT MONITORING TO CW_HANADB_EXPORTER_ROLE;
GRANT CW_HANADB_EXPORTER_ROLE TO CW_HANADB_EXPORTER_USER;
```

- Python 3.8 — Installieren Sie Python 3.8 oder neuere Versionen auf Ihrem Betriebssystem. Verwenden Sie die neueste Version von Python. Wenn Python3 auf Ihrem Betriebssystem nicht erkannt wird, wird Python 3.6 installiert.

  Weitere Informationen hierzu finden Sie unter installation example.

  > **Note**
  >
  > Die manuelle Installation von Python 3.8 oder höher ist für SuSE Linux 15 SP4, RedHat Linux 8.6 und neuere Betriebssysteme erforderlich.

- Pip3 — Installieren Sie das Installationsprogramm pip3 auf Ihrem Betriebssystem. Wenn pip3 auf Ihrem Betriebssystem nicht erkannt wird, wird es installiert.
- hdbclient — CloudWatch Application Insights verwendet den Python-Treiber, um eine Verbindung zur SAP HANA-Datenbank herzustellen. Wenn der Client nicht unter Python3 installiert ist, stellen Sie sicher, dass Sie die Hdbclient-Tar-Dateiversion unter haben. `2.10 or later` `/hana/shared/SID/hdbclient/`
- CloudWatch Amazon-Agent — Stellen Sie sicher, dass Sie keinen bereits vorhandenen CloudWatch Agenten auf Ihrer Amazon EC2-Instance ausführen. Wenn Sie einen CloudWatch Agenten installiert haben, stellen Sie sicher, dass Sie die Konfiguration der Ressourcen, die Sie in CloudWatch Application Insights verwenden, aus der vorhandenen CloudWatch Agenten-Konfigurationsdatei entfernen, um einen Zusammenführungskonflikt zu vermeiden. Weitere Informationen finden Sie unter Erstellen oder bearbeiten Sie die CloudWatch Agenten-Konfigurationsdatei manuell.
- AWS Systems Manager Manager-Aktivierung — Installieren Sie den SSM-Agent auf Ihren Instances, und die Instanzen müssen für SSM aktiviert sein. Informationen zur Installation des SSM-Agenten finden Sie unter Arbeiten mit SSM Agent im AWS Systems Manager Manager-Benutzerhandbuch.

- Rollen für Amazon-EC2-Instances – Sie müssen die folgenden Amazon-EC2-Instance-Rollen anhängen, um Ihre Datenbank zu konfigurieren.
  - Sie müssen die `AmazonSSMManagedInstanceCore`-Rolle anfügen, um Systems Manager zu aktivieren. Weitere Informationen finden Sie unter Beispiele für identitätsbasierte AWS Systems Manager -Richtlinien.
  - Sie müssen das anhängen`CloudWatchAgentServerPolicy`, damit Instanzmetriken und -protokolle ausgegeben werden können. CloudWatch Weitere Informationen finden Sie unter Erstellen von IAM-Rollen und -Benutzern zur Verwendung mit CloudWatch Agenten.
  - Sie müssen die folgende IAM-Inline-Richtlinie an die Amazon-EC2-Instancerolle anhängen, um das in AWS Secrets Manager gespeicherte Passwort auszulesen. Weitere Informationen zu Inline-Richtlinien finden Sie unter Inline-Richtlinien im AWS Identity and Access Management -Benutzerhandbuch.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
        }
    ]
}
```

- AWS Ressourcengruppen — Sie müssen eine Ressourcengruppe erstellen, die alle zugehörigen AWS Ressourcen umfasst, die von Ihrem Anwendungsstapel verwendet werden, um Ihre Anwendungen in CloudWatch Application Insights zu integrieren. Dies umfasst Amazon-EC2-Instances und Amazon-EBS-Volumes, auf denen Ihre SAP HANA-Datenbank ausgeführt wird. Wenn es mehrere Datenbanken pro Konto gibt, empfehlen wir, dass Sie eine Ressourcengruppe erstellen, die die AWS Ressourcen für jedes SAP HANA-Datenbanksystem enthält.
- IAM-Berechtigungen – Für Nicht-Admin-Benutzer:
  - Sie müssen eine AWS Identity and Access Management (IAM-) Richtlinie erstellen, die es Application Insights ermöglicht, eine dienstbezogene Rolle zu erstellen, und sie Ihrer Benutzeridentität zuordnen. Wie Sie die Richtlinie anfügen, erfahren Sie unter IAM-Richtlinie.

- Der Benutzer muss berechtigt sein, ein Geheimnis zu erstellen, in AWS Secrets Manager dem die Anmeldeinformationen des Datenbankbenutzers gespeichert werden. Weitere Informationen finden Sie unter Beispiel: Berechtigung zum Erstellen von Secrets.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:CreateSecret"
      ],
      "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
    }
  ]
}
```

- Dienstverknüpfte Rolle — Application Insights verwendet AWS Identity and Access Management (IAM) dienstbezogene Rollen. Es wird für Sie eine serviceverknüpfte Rolle erstellt, wenn Sie Ihre erste Application-Insights-Anwendung in der Application-Insights-Konsole erstellen. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für CloudWatch Application Insights.

Einrichten der SAP HANA-Datenbank für die Überwachung

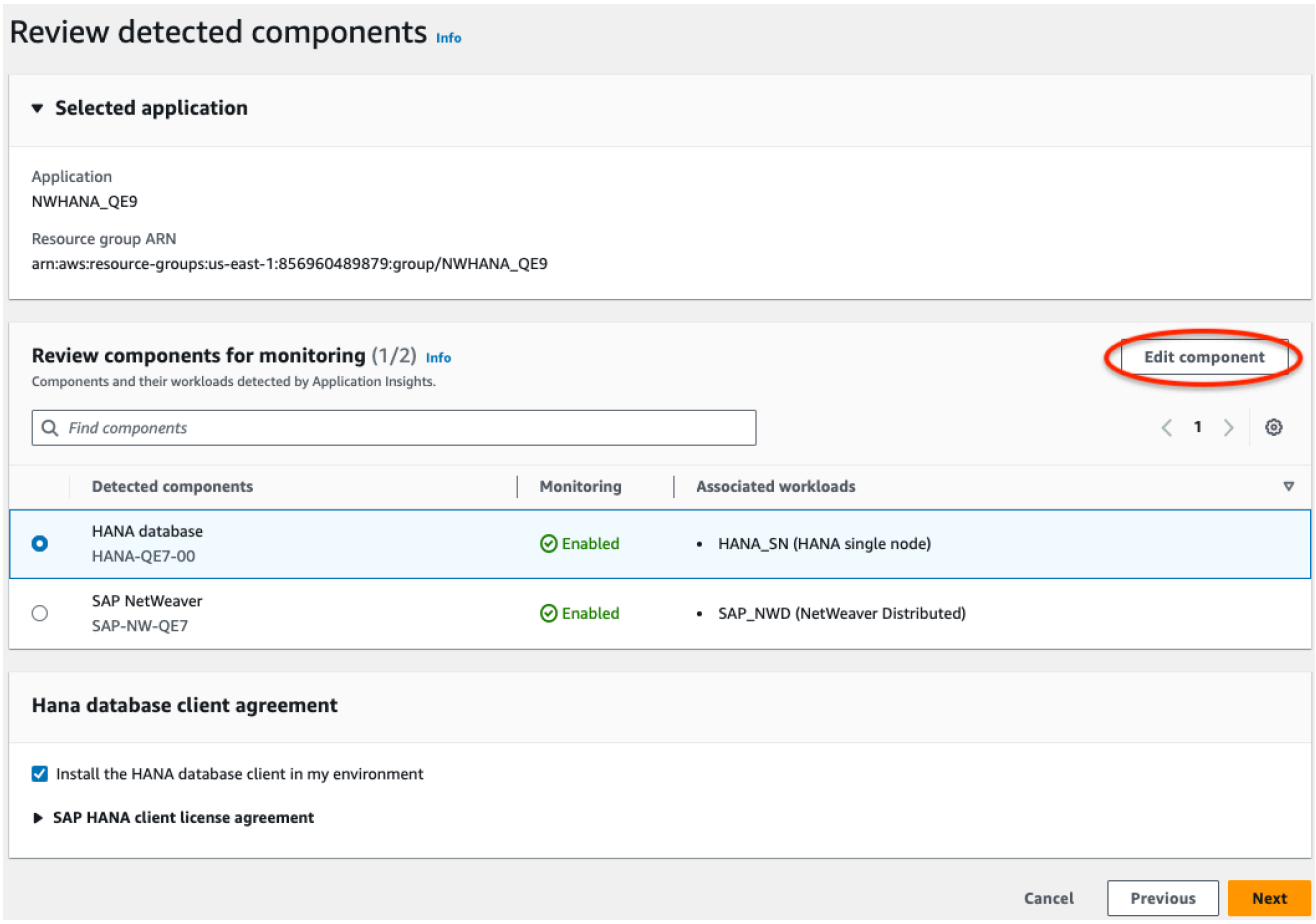
Gehen Sie wie folgt vor, um die Überwachung für Ihre SAP HANA-Datenbank einzurichten:

1. Öffnen Sie die CloudWatch-Konsole.
2. Wählen Sie im linken Navigationsbereich unter Insights die Option Application Insights aus.
3. Auf der Seite Application Insights sehen Sie die Anwendungen, die mit Application Insights überwacht werden, samt Überwachungsstatus. Wählen Sie in der oberen rechten Ecke Add an application (Eine Anwendung hinzufügen) aus.
4. Auf der Seite Specify application details (Anwendungsdetails festlegen) wählen Sie in der Dropdown-Liste unter Resource group die AWS -Ressourcengruppe aus, die Ihre SAP HANA-Datenbankressourcen enthält. Wenn Sie keine Ressourcengruppe für Ihre Anwendung erstellt haben, können Sie das mit Create new resource group (Neue Ressourcengruppe erstellen) im Dropdown-Menü Resource group (Ressourcengruppe) tun. Weitere Informationen zu Ressourcengruppen finden Sie im AWS -Benutzerhandbuch zu Resource Groups.

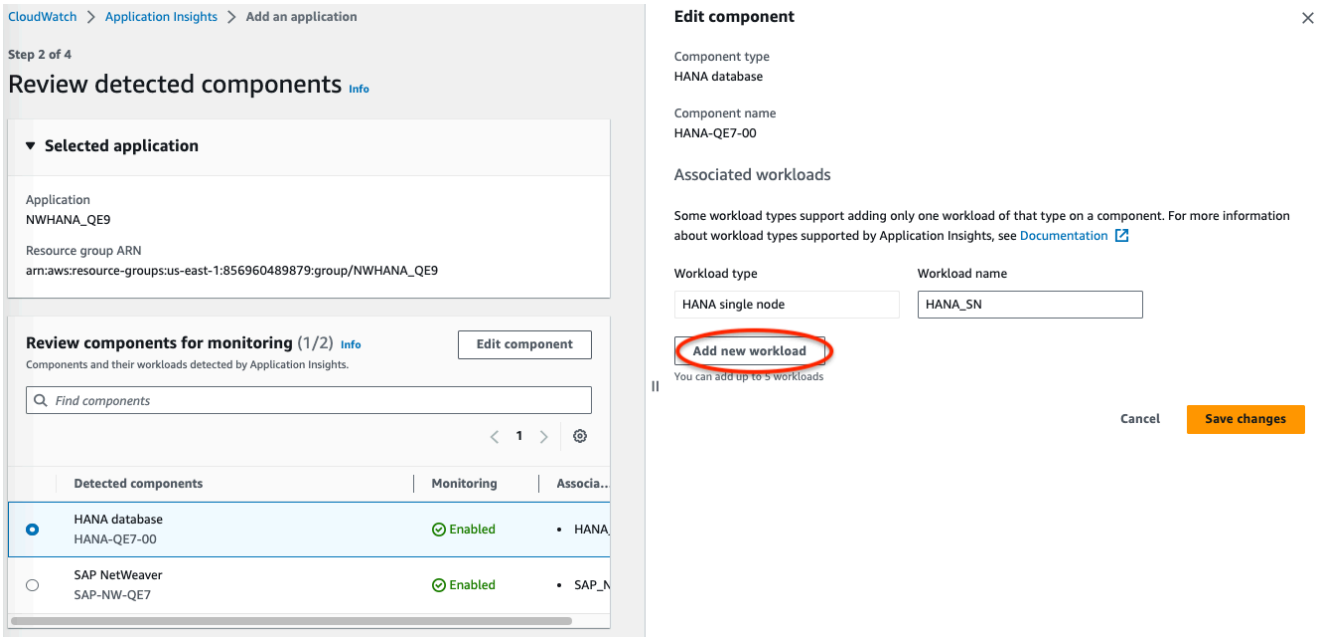
5. Aktivieren Sie unter CloudWatch Ereignisse überwachen das Kontrollkästchen, um die Überwachung von Application Insights mit CloudWatch Ereignissen zu integrieren, um Erkenntnisse aus Amazon EBS, Amazon EC2, Amazon ECS AWS CodeDeploy, AWS Health APIs und Benachrichtigungen, Amazon RDS, Amazon S3 und zu erhalten. AWS Step Functions
6. Aktivieren Sie unter Integrieren mit AWS Systems Manager OpsCenter das Kontrollkästchen neben AWS Systems Manager OpsCenter OpsItems Für Abhilfemaßnahmen generieren, um Benachrichtigungen anzuzeigen und zu erhalten, wenn bei den ausgewählten Anwendungen Probleme festgestellt werden. Geben Sie einen ARN für das SNS-Thema an, um die Operationen nachzuverfolgen OpsItems, die zur Lösung von so genannten operativen Arbeitsaufgaben ausgeführt werden, die sich auf Ihre AWS Ressourcen beziehen.
7. Sie können optional Tags eingeben, die Ihnen helfen, Ihre Ressourcen zu identifizieren und zu organisieren. CloudWatch Application Insights unterstützt sowohl tagbasierte als auch AWS CloudFormation stapelbasierte Ressourcengruppen, mit Ausnahme von Gruppen. Application Auto Scaling Weitere Informationen finden Sie unter Tag-Editor im Benutzerhandbuch zu AWS Resource Groups und Tags.
8. Wählen Sie Next (Weiter) aus, um mit der Einrichtung der Überwachung fortzufahren.
9. Auf der Seite Entdeckte Komponenten überprüfen werden die überwachten Komponenten und ihre Workloads aufgeführt, die von CloudWatch Application Insights automatisch erkannt wurden.

    a. Um Workloads zu einer Komponente hinzuzufügen, die einen erkannten SAP-HANA-Einzelknoten-Workload enthält, markieren Sie die Komponente und wählen dann Komponente bearbeiten.

    > **Note**
    >
    > Komponenten, die einen erkannten SAP-HANA-Multiknoten- oder HANA-High-Availability-Workload enthalten, unterstützen nur einen Workload auf einer Komponente.



b. Um einen neuen Workload hinzuzufügen, wählen Sie Neuen Workload hinzufügen.



c. Wenn Sie mit der Bearbeitung der Workloads fertig sind, wählen Sie Änderungen speichern.

10. Wählen Sie Weiter aus.
11. Geben Sie auf der Seite Komponentendetails angeben den Benutzernamen und das Passwort ein.
12. Überprüfen Sie die Konfiguration der Anwendungsüberwachung und wählen Sie Submit (Absenden) aus.
13. Es öffnet sich die Seite mit den Anwendungsdetails, auf der Sie die Anwendungsübersicht, die Liste der überwachten Komponenten und Workloads sowie der nicht überwachten Komponenten und Workloads sehen können. Wenn Sie das Optionsfeld neben einer Komponente oder einem Workload auswählen, können Sie auch den Konfigurationsverlauf, die Protokollmuster und alle von Ihnen erstellten Tags anzeigen. Wenn Sie Ihre Konfiguration einreichen, stellt Ihr Konto alle Metriken und Alarme für Ihr SAP HANA-System bereit, was bis zu 2 Stunden dauern kann.

Verwalten der Überwachung der SAP HANA-Datenbank

Sie können Benutzeranmeldeinformationen, Metriken und Protokollpfade für Ihre SAP HANA-Datenbank verwalten, indem Sie die folgenden Schritte ausführen:

1. Öffnen Sie die CloudWatch-Konsole.
2. Wählen Sie im linken Navigationsbereich unter Insights die Option Application Insights aus.
3. Auf der Seite Application Insights sehen Sie die Anwendungen, die mit Application Insights überwacht werden, samt Überwachungsstatus.
4. Unter Monitored components (Überwachte Komponenten) wählen Sie das Optionsfeld neben dem Komponentennamen aus. Wählen Sie dann Manage monitoring (Überwachung verwalten) aus.
5. Unter EC2 instance group logs (Protokolle für EC2-Instance-Gruppen) können Sie den vorhandenen Protokollpfad, den Protokollmustersatz und den Namen der Protokollgruppe bearbeiten. Darüber hinaus können Sie bis zu drei zusätzliche Application logs (Anwendungsprotokolle) hinzufügen.
6. Unter Metrics (Metriken) können Sie die SAP HANA-Metriken nach Ihren Anforderungen auswählen. SAP HANA-Metriknamen haben das Präfix `hanadb`. Sie können bis zu 40 Metriken pro Komponente hinzufügen.
7. Geben Sie unter HANA configuration (HANA-Konfiguration) das Passwort und den Benutzernamen für die SAP HANA-Datenbank ein. Dies sind der Benutzername und das

Passwort, die der CloudWatch Amazon-Agent verwendet, um sich mit der SAP HANA-Datenbank zu verbinden.

8. Unter Benutzerdefinierte Alarme können Sie zusätzliche Alarme hinzufügen, die von CloudWatch Application Insights überwacht werden sollen.
9. Überprüfen Sie die Konfiguration der Anwendungsüberwachung und wählen Sie Submit (Absenden) aus. Wenn Sie Ihre Konfiguration einreichen, aktualisiert Ihr Konto alle Metriken und Alarme für Ihr SAP HANA-System, was bis zu 2 Stunden dauern kann.

Von CloudWatch Application Insights erkannte SAP HANA-Probleme anzeigen und beheben

Die folgenden Abschnitte enthalten Schritte, mit denen Sie häufig auftretende Fehler beheben können, die bei der Konfiguration der Überwachung für SAP HANA in Application Insights auftreten.

Themen zur Fehlerbehebung

- SAP HANA-Datenbank erreicht das Limit für die Speicherzuweisung
- Der Datenträger ist voll
- Das SAP HANA-Backup läuft nicht mehr

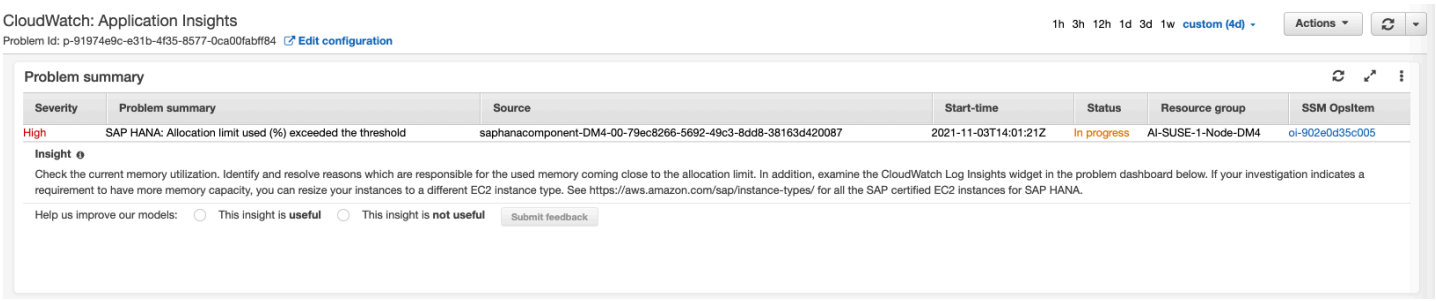
SAP HANA-Datenbank erreicht das Limit für die Speicherzuweisung

Beschreibung

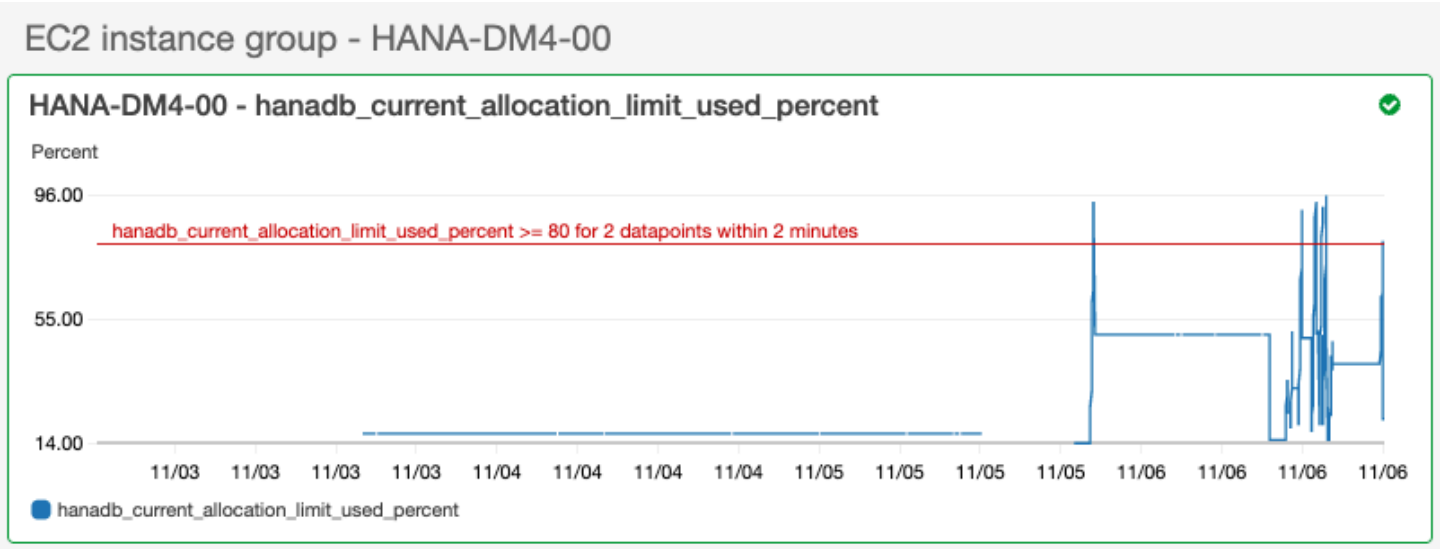
Ihre SAP-Anwendung, die durch eine SAP HANA-Datenbank unterstützt wird, funktioniert aufgrund des hohen Speicherdrucks nicht korrekt, was zu einer Leistungsverschlechterung der Anwendung führt.

Auflösung

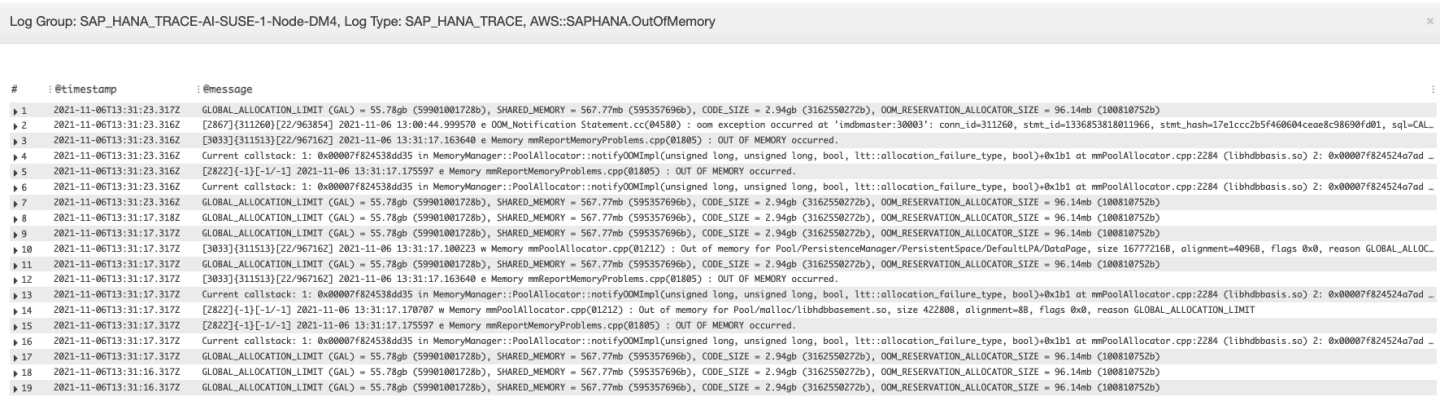
Sie können herausfinden, welche Anwendungsebene das Problem verursacht, indem Sie auf dem dynamischen Dashboard nachsehen. Dort sind die zugehörigen Metriken und Protokolldateiausschnitte angezeigt. Im folgenden Beispiel könnte das Problem auf eine große Datenlast im SAP HANA-System zurückzuführen sein.



Die Speicherzuweisung überschreitet den Schwellenwert von 80 Prozent des gesamten Speicherzuweisungslimits.



Die Protokollgruppe meldet, dass das Schema `BNR-DATA` und die Tabelle `IMDBMASTER_30003` nicht mehr ausreichend Speicher haben. Darüber hinaus zeigt die Protokollgruppe den genauen Zeitpunkt, das aktuelle globale Standortlimit, den gemeinsam genutzten Speicher, die Codegröße und die Größe der OOM-Reservierungszuweisung an.



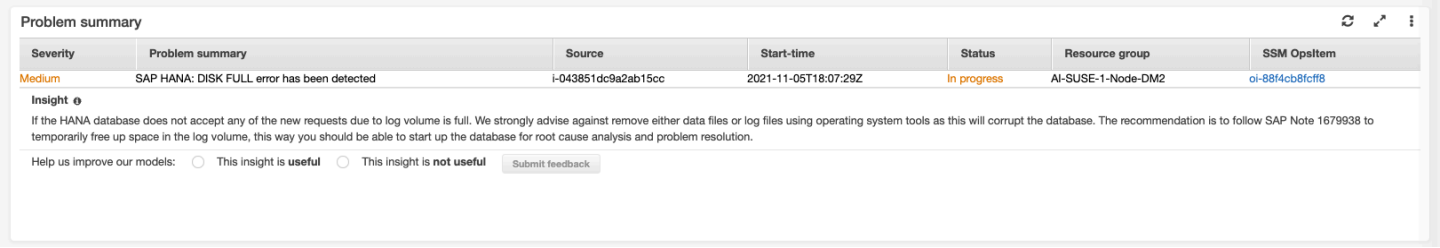
Der Datenträger ist voll

Beschreibung

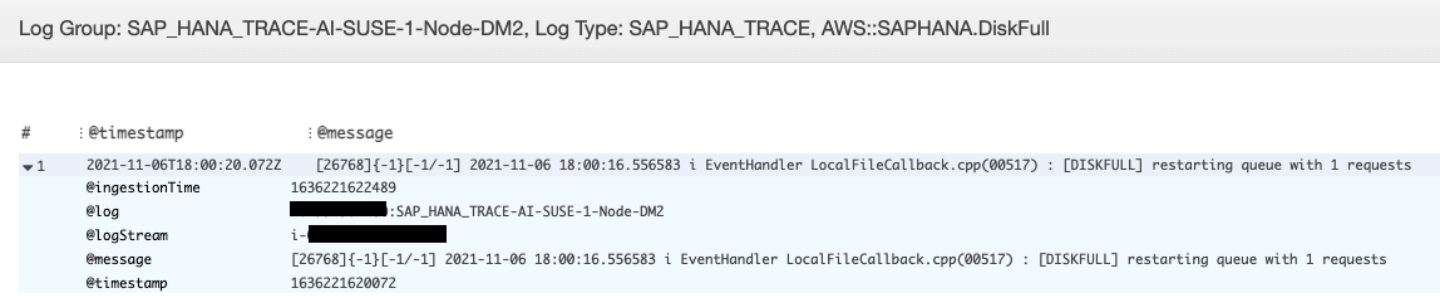
Die SAP-Anwendung, die von einer SAP HANA-Datenbank unterstützt wird, reagiert nicht mehr, was dazu führt, dass nicht auf die Datenbank zugegriffen werden kann.

Auflösung

Sie können herausfinden, welche Datenbankebene das Problem verursacht, indem Sie auf dem dynamischen Dashboard nachsehen. Dort sind die zugehörigen Metriken und Protokolldateiausschnitte angezeigt. Im folgenden Beispiel könnte das Problem darin bestehen, dass der Administrator die automatische Protokollsicherung nicht aktiviert hat, was dazu führt, dass das Verzeichnis sap/hana/log überfüllt wurde.



Das Protokollgruppen-Widget im Problem-Dashboard zeigt das Ereignis `DISKFULL`.



Das SAP HANA-Backup läuft nicht mehr

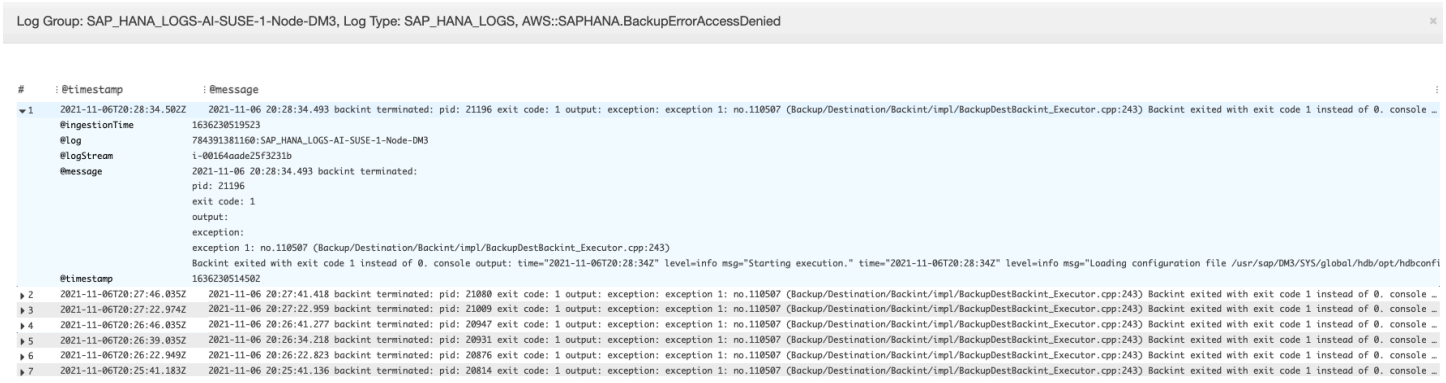
Beschreibung

Die SAP-Anwendung, die von einer SAP HANA-Datenbank unterstützt wird, funktioniert nicht mehr.

Auflösung

Sie können herausfinden, welche Datenbankebene das Problem verursacht, indem Sie auf dem dynamischen Dashboard nachsehen. Dort sind die zugehörigen Metriken und Protokolldateiausschnitte angezeigt.

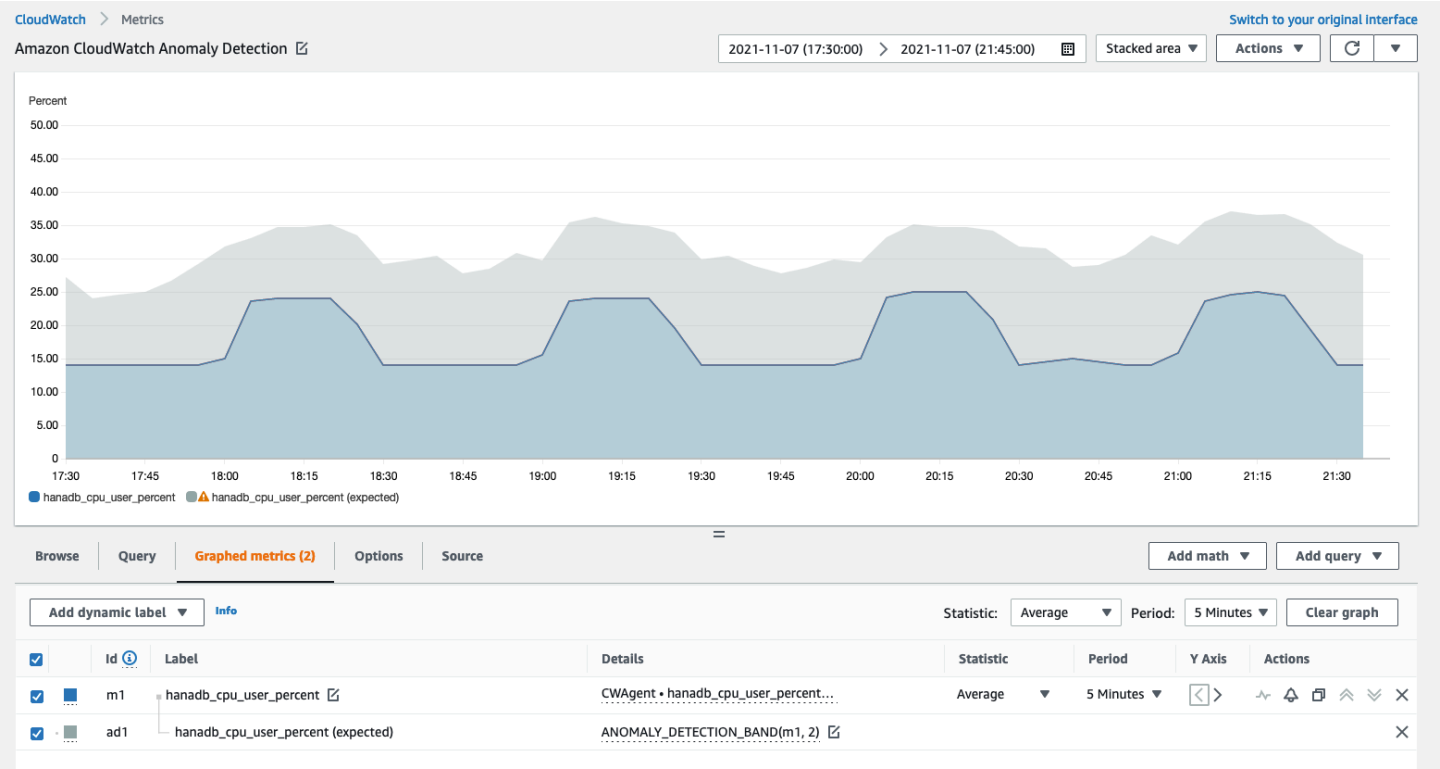
Das Protokollgruppen-Widget im Problem-Dashboard zeigt das Ereignis `ACCESS DENIED`. Dies umfasst zusätzliche Informationen wie den S3 Bucket, den S3 Bucket-Ordner und die S3 Bucket-Region.



Erkennung von Anomalien für SAP HANA

Bei bestimmten SAP HANA-Metriken, wie z. B. der Anzahl der Threads, werden statistische Algorithmen und Algorithmen CloudWatch für maschinelles Lernen verwendet, um den Schwellenwert zu definieren. Diese Algorithmen analysieren kontinuierlich Metriken der SAP HANA-Datenbank, ermitteln normale Baseline-Werte und zeigen Anomalien an, wobei nur minimale Benutzereingriffe erforderlich sind. Die Algorithmen generieren ein Modell zur Erkennung von Anomalien, das einen Bereich von erwartbaren Werten ermittelt, wie sie bei einem normalen Metrikverhalten auftreten würden.

Anomalieerkennungsalgorithmen berücksichtigen saisonale und trendbasierte Änderungen von Metriken. Die saisonalen Änderungen können stündlich, täglich oder wöchentlich erfolgen, wie in den folgenden Beispielen für die CPU-Auslastung durch SAP HANA gezeigt.



Nachdem Sie ein Modell erstellt haben, bewertet die CloudWatch Anomalieerkennung das Modell kontinuierlich und nimmt Anpassungen vor, um sicherzustellen, dass es so genau wie möglich ist. Dies beinhaltet das Umtraining des Modells für den Fall, dass sich die Metrikwerte im Laufe der Zeit weiterentwickeln oder plötzlich ändern. Außerdem gibt es Prädiktoren zur Verbesserung der Modelle für Metriken, die saisonal, sehr hoch oder zerstreut sind.

Fehlerbehebung bei Application Insights für SAP HANA

Dieser Abschnitt enthält Schritte, mit denen Sie häufige Fehler beheben können, die vom Application-Insights-Dashboard zurückgegeben werden.

Es konnten nicht mehr als 60 überwachte Metriken hinzugefügt werden

Die Ausgabe zeigt den folgenden Fehler.

```
Component cannot have more than 60 monitored metrics
```

Hauptursache — Das aktuelle Metriklimit liegt bei 60 überwachten Metriken pro Komponente.

Lösung — Um unter dem Limit zu bleiben, entfernen Sie Metriken, die nicht erforderlich sind.

Nach dem Onboarding-Prozess werden keine SAP Metriken angezeigt

Finden Sie anhand der folgenden Informationen heraus, warum SAP-Metriken nach dem Onboarding-Prozess nicht im Dashboard angezeigt werden. Der erste Schritt besteht darin, anhand der AWS Management Console oder Exporter-Protokolle einer Amazon EC2 EC2-Instance zu beheben, warum die SAP-Metriken nicht angezeigt werden. Überprüfen Sie als Nächstes die Fehlerausgabe, um eine Lösung zu finden.

Beheben Sie, warum SAP-Metriken nach dem Onboarding nicht angezeigt werden

Sie können die AWS Management Console oder Exporter-Protokolle einer Amazon EC2 EC2-Instance zur Fehlerbehebung verwenden.

AWS Management Console

Fehlerbehebung Nach dem Onboarding über die Konsole werden keine SAP-Metriken angezeigt

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im linken Navigationsbereich State Manager aus.
3. Überprüfen Sie unter Verknüpfungen den Status des Dokuments`AWSEC2-ApplicationInsightsCloudwatchAgentInstallAndConfigure`. Wenn der Status lautet`Failed`, wählen Sie unter Ausführungs-ID die fehlgeschlagene ID aus und sehen Sie sich die Ausgabe an.
4. Überprüfen Sie unter Verknüpfungen den Status des Dokuments`AWS-ConfigureAWSPackage`. Wenn der Status lautet`Failed`, wählen Sie unter Ausführungs-ID die fehlgeschlagene ID aus und sehen Sie sich die Ausgabe an.

Exporter logs from Amazon EC2 instance

Fehlerbehebung: Nach dem Onboarding werden mithilfe von Exportprotokollen keine SAP-Metriken angezeigt

1. Connect zu der Amazon EC2 EC2-Instance her, auf der Ihre SAP HANA-Datenbank läuft.
2. Finden Sie die richtige Benennungskonvention für die `WORKLOAD_SHORT_NAME` Verwendung des folgenden Befehls. Sie werden diesen Kurznamen in den folgenden beiden Schritten verwenden.

```
sudo systemctl | grep exporter
```

> **Note**
>
> Application Insights fügt dem Dienstnamen je nach ausgeführtem Workload ein Suffix hinzu. `WORKLOAD_SHORT_NAME` Die Kurznamen für SAP HANA-Bereitstellungen mit einem Knoten, mehreren Knoten und Hochverfügbarkeitsbereitstellungen lauten `HANA_SN`HANA_MN, und. `HANA_HA`

3. Um nach Fehlern in den Serviceprotokollen des Exporter Managers zu suchen, führen Sie den folgenden Befehl aus und `WORKLOAD_SHORT_NAME` ersetzen Sie ihn durch den Kurznamen, den Sie in gefunden haben. Step 2

```
sudo journalctl -e --unit=prometheus-
hanadb_exporter_manager_WORKLOAD_SHORT_NAME.service
```

4. Wenn in den Dienstprotokollen des Export-Managers kein Fehler angezeigt wird, suchen Sie nach Fehlern in den Serviceprotokollen des Exporter, indem Sie den folgenden Befehl ausführen.

```
sudo journalctl -e --unit=prometheus-hanadb_exporter_WORKLOAD_SHORT_NAME.service
```

Behebung der häufigsten Ursachen dafür, dass SAP-Metriken nach dem Onboarding nicht angezeigt werden

In den folgenden Beispielen wird beschrieben, wie die häufigsten Ursachen dafür behoben werden können, dass SAP-Metriken nach dem Onboarding nicht angezeigt werden.

- Die Ausgabe zeigt den folgenden Fehler.

```
Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/amazon-
cloudwatch-agent.d/default ...
Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/
amazon-cloudwatch-agent.d/ssm_AmazonCloudWatch-ApplicationInsights-
SSMParameterForTESTCWEC2INSTANCEi0d88867f1f3e36285.tmp ...
2023/11/30 22:25:17 Failed to merge multiple json config files.
2023/11/30 22:25:17 Failed to merge multiple json config files.
```

```
2023/11/30 22:25:17 Under path : /metrics/append_dimensions | Error : Different
 values are specified for append_dimensions
2023/11/30 22:25:17 Under path : /metrics/metrics_collected/disk | Error : Different
 values are specified for disk
2023/11/30 22:25:17 Under path : /metrics/metrics_collected/mem | Error : Different
 values are specified for mem
2023/11/30 22:25:17 Configuration validation first phase failed. Agent version: 1.0.
 Verify the JSON input is only using features supported by this version.
```

Lösung — Application Insights versucht, dieselben Metriken zu konfigurieren, die als Teil der vorhandenen CloudWatch Agentenkonfigurationsdatei vorkonfiguriert sind. Entfernen Sie die vorhandenen Dateien unter `/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/` oder entfernen Sie die Metriken, die den Konflikt verursachen, aus der vorhandenen CloudWatch Agentenkonfigurationsdatei.

- Die Ausgabe zeigt den folgenden Fehler.

```
Unable to find a host with system database, for more info rerun using -v
```

Lösung — Der Benutzername, das Passwort oder der Datenbankport sind möglicherweise falsch. Stellen Sie sicher, dass der Benutzername, das Passwort und der Port gültig sind, und führen Sie dann den Onboarding-Prozess erneut aus.

- Die Ausgabe zeigt den folgenden Fehler.

```
This hdbcli installer is not compatible with your Python interpreter
```

Lösung — Aktualisieren Sie pip3 und wheel, wie im folgenden Beispiel für Python 3.6 gezeigt.

```
python3.6 -m pip install --upgrade pip setuptools wheel
```

- Die Ausgabe zeigt den folgenden Fehler.

```
Unable to install hdbcli using pip3. Please try to install it
```

Lösung — Stellen Sie sicher, dass Sie die `hdbclient` Voraussetzungen erfüllt haben, oder installieren Sie `hdbclient` manuell unter pip3.

- Die Ausgabe zeigt den folgenden Fehler.

```
Package 'boto3' requires a different Python: 3.6.15 not in '>= 3.7'
```

Auflösung — Python 3.8 oder höher ist für diese Betriebssystemversion erforderlich. Überprüfen Sie die Voraussetzungen für Python 3.8 und installieren Sie es.

- Die Ausgabe zeigt einen der folgenden Installationsfehler.

```
Can not execute `setup.py` since setuptools is not available in the build environment
```

or

```
[SSL: CERTIFICATE_VERIFY_FAILED]
```

Lösung — Installieren Sie Python mithilfe von SUSE Linux-Befehlen, wie im folgenden Beispiel gezeigt. Das folgende Beispiel installiert die neueste Version von Python 3.8.

```
wget https://www.python.org/ftp/python/3.8.<LATEST_RELEASE>/
Python-3.8.<LATEST_RELEASE>.tgz
tar xf Python-3.*
cd Python-3.*/
sudo zypper install make gcc-c++ gcc automake autoconf libtool
sudo zypper install zlib-devel
sudo zypper install libopenssl-devel libffi-devel
./configure --with-ensurepip=install
sudo make
sudo make install
sudo su
python3.8 -m pip install --upgrade pip setuptools wheel
```

Tutorial: Monitoring für SAP einrichten NetWeaver

Dieses Tutorial zeigt, wie Amazon CloudWatch Application Insights konfiguriert wird, um die Überwachung für SAP einzurichten NetWeaver. Sie können die automatischen Dashboards von CloudWatch Application Insights verwenden, um Problemdetails zu visualisieren, die Fehlerbehebung zu beschleunigen und die mittlere Zeit bis zur Lösung (MTTR) für Ihre NetWeaver SAP-Anwendungsserver zu reduzieren.

CloudWatch Application Insights für SAP-Themen NetWeaver

Unterstützte Umgebungen

CloudWatch Application Insights unterstützt die Bereitstellung von AWS Ressourcen für die folgenden Systeme und Muster.

- Bereitstellung des NetWeaver SAP-Standardsystems.
- SAP NetWeaver Distributed Deployments auf mehreren Amazon EC2 EC2-Instances.
- AZ-übergreifendes NetWeaver SAP-Hochverfügbarkeits-Setup — SAP NetWeaver mit Hochverfügbarkeit, konfiguriert für zwei Availability Zones mithilfe von SUSE/RHEL-Clustering.

Unterstützte Betriebssysteme

CloudWatch Application Insights for SAP NetWeaver wird auf den folgenden Betriebssystemen unterstützt:

- Oracle Linux 8
- Red Hat Enterprise Linux 7.6
- Red Hat Enterprise Linux 7.7
- Red Hat Enterprise Linux 7.9
- Red Hat Enterprise Linux 8.1
- Red Hat Enterprise Linux 8.2
- Red Hat Enterprise Linux 8.4

- Red Hat Enterprise Linux 8.6
- SUSE Linux Enterprise Server 15 für SAP
- SUSE Linux Enterprise Server 15 SP1 für SAP
- SUSE Linux Enterprise Server 15 SP2 für SAP
- SUSE Linux Enterprise Server 15 SP3 für SAP
- SUSE Linux Enterprise Server 15 SP4 für SAP
- SUSE Linux Enterprise Server 12 SP4 für SAP
- SUSE Linux Enterprise Server 12 SP5 für SAP
- SUSE Linux Enterprise Server 15 außer Hochverfügbarkeitsmustern
- SUSE Linux Enterprise Server 15 SP1 außer Hochverfügbarkeitsmustern
- SUSE Linux Enterprise Server 15 SP2 außer Hochverfügbarkeitsmustern
- SUSE Linux Enterprise Server 15 SP3 außer Hochverfügbarkeitsmustern
- SUSE Linux Enterprise Server 15 SP4 außer Hochverfügbarkeitsmustern
- SUSE Linux Enterprise Server 12 SP4 außer Hochverfügbarkeitsmustern
- SUSE Linux Enterprise Server 12 SP5 außer Hochverfügbarkeitsmustern

Features

CloudWatch Application Insights for SAP NetWeaver 7.0x—7.5x (einschließlich ABAP Platform) bietet die folgenden Funktionen:

- NetWeaver Automatische Erkennung von SAP-Workloads
- Automatische NetWeaver SAP-Alarmerstellung auf der Grundlage statischer Schwellenwerte
- Automatische Erkennung von NetWeaver SAP-Protokollmustern
- Gesundheits-Dashboard für SAP NetWeaver
- Problem-Dashboard für SAP NetWeaver

Voraussetzungen

Sie müssen die folgenden Voraussetzungen erfüllen, um SAP NetWeaver mit CloudWatch Application Insights zu konfigurieren:

- AWS Systems Manager Manager-Aktivierung — Installieren Sie den SSM-Agent auf Ihren Amazon EC2 EC2-Instances und aktivieren Sie die Instances für SSM. Informationen zur Installation des SSM-Agent finden Sie unter Einrichten von AWS Systems Manager im AWS -Systems-Manager-Benutzerhandbuch.
- Amazon EC2 EC2-Instance-Rollen — Sie müssen die folgenden Amazon EC2 EC2-Instance-Rollen anhängen, um Ihre NetWeaver SAP-Überwachung zu konfigurieren.
  - Sie müssen die `AmazonSSMManagedInstanceCore`-Rolle anfügen, um Systems Manager zu aktivieren. Weitere Informationen finden Sie unter Beispiele für identitätsbasierte AWS Systems Manager -Richtlinien.
  - Sie müssen die `CloudWatchAgentServerPolicy` Richtlinie anhängen, damit Instance-Metriken und -Protokolle ausgegeben werden können. CloudWatch Weitere Informationen finden Sie unter IAM-Rollen und -Benutzer für die Verwendung mit CloudWatch Agenten erstellen.
- AWS Ressourcengruppen — Sie müssen eine Ressourcengruppe erstellen, die alle zugehörigen AWS Ressourcen umfasst, die von Ihrem Anwendungsstapel verwendet werden, um Ihre Anwendungen in CloudWatch Application Insights zu integrieren. Dazu gehören Amazon EC2 EC2-Instances, Amazon EFS und Amazon EBS-Volumes, auf denen Ihre NetWeaver SAP-Anwendungsserver ausgeführt werden. Wenn es mehrere NetWeaver SAP-Systeme pro Konto gibt, empfehlen wir, dass Sie eine Ressourcengruppe erstellen, die die AWS Ressourcen für jedes SAP-System NetWeaver enthält. Weitere Informationen zu Ressourcengruppen finden Sie im AWS -Benutzerhandbuch zu Ressourcengruppen und Tags.
- IAM-Berechtigungen — Für Benutzer, die keinen Administratorzugriff haben, müssen Sie eine AWS Identity and Access Management (IAM-) Richtlinie erstellen, die es Application Insights ermöglicht, eine dienstbezogene Rolle zu erstellen und sie der Benutzeridentität zuzuordnen. Informationen dazu, wie Sie eine IAM-Richtlinie erstellen, finden Sie unter IAM-Richtlinie.
- Dienstverknüpfte Rolle — Application Insights verwendet AWS Identity and Access Management (IAM) dienstbezogene Rollen. Es wird für Sie eine serviceverknüpfte Rolle erstellt, wenn Sie Ihre erste Application-Insights-Anwendung in der Application-Insights-Konsole erstellen. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für CloudWatch Application Insights.
- CloudWatch Amazon-Agent — Application Insights installiert und konfiguriert den CloudWatch Agenten. Wenn Sie den CloudWatch Agenten installiert haben, behält Application Insights Ihre Konfiguration bei. Um einen Zusammenführungskonflikt zu vermeiden, entfernen Sie die Konfiguration der Ressourcen, die Sie in Application Insights verwenden möchten, aus der vorhandenen CloudWatch Agentenkonfigurationsdatei. Weitere Informationen finden Sie unter Erstellen oder bearbeiten Sie die CloudWatch Agenten-Konfigurationsdatei manuell.

Richten Sie Ihre NetWeaver SAP-Anwendungsserver für die Überwachung ein

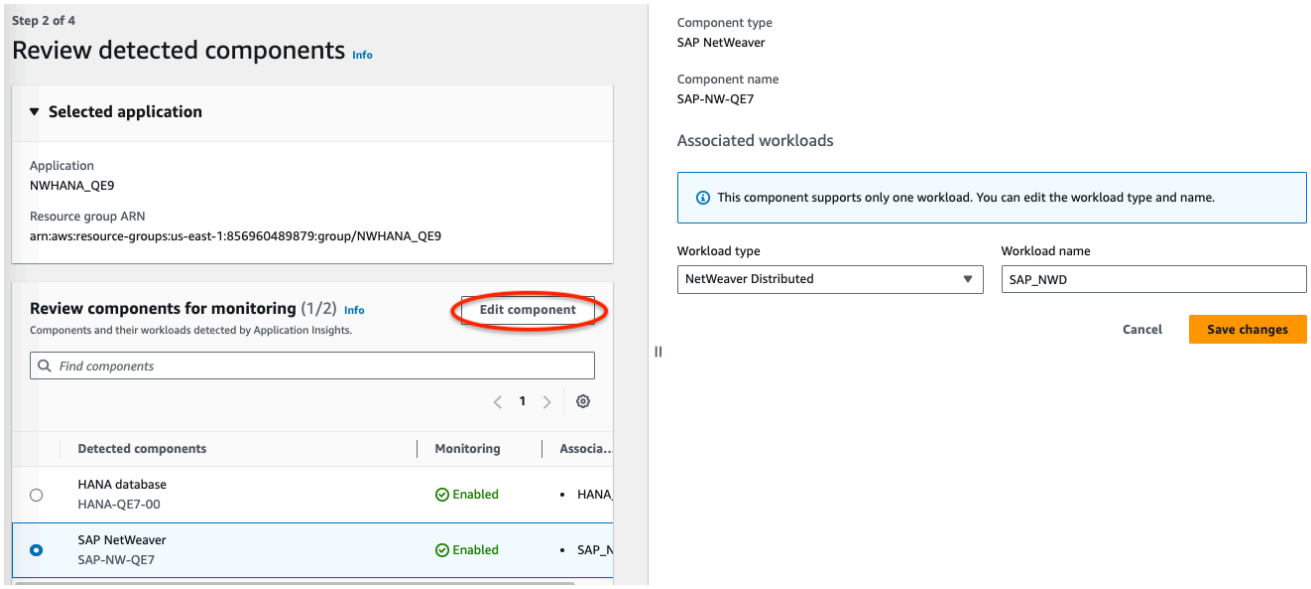
Gehen Sie wie folgt vor, um die Überwachung für Ihre NetWeaver SAP-Anwendungsserver einzurichten.

So richten Sie die Überwachung ein

1. Öffnen Sie die CloudWatch -Konsole.
2. Wählen Sie im linken Navigationsbereich unter Insights (Einblicke) die Option Application Insights (Anwendungseinblicke) aus.
3. Auf der Seite Application Insights sehen Sie die Anwendungen, die mit Application Insights überwacht werden, samt Überwachungsstatus. Wählen Sie in der oberen rechten Ecke Add an application (Eine Anwendung hinzufügen) aus.
4. Wählen Sie auf der Seite Anwendungsdetails angeben aus der Dropdownliste unter Ressourcengruppe die Ressourcengruppe aus, die AWS Sie erstellt haben und die Ihre NetWeaver SAP-Ressourcen enthält. Wenn Sie keine Ressourcengruppe für Ihre Anwendung erstellt haben, können Sie das mit Create new resource group (Neue Ressourcengruppe erstellen) in der Dropdown-Liste Resource group (Ressourcengruppe) tun.
5. Aktivieren Sie unter Automatic monitoring of new resources (Automatische Überwachung neuer Ressourcen) das Kontrollkästchen, damit Application Insights die Ressourcen, die nach dem Einbinden der Ressourcengruppe der Anwendung hinzugefügt werden, automatisch überwacht.
6. Aktivieren Sie unter EventBridge Ereignisse überwachen das Kontrollkästchen, um die Überwachung von Application Insights mit CloudWatch Ereignissen zu integrieren, um Einblicke von Amazon EBS, Amazon EC2, Amazon ECS AWS CodeDeploy, AWS Health APIs und Benachrichtigungen, Amazon RDS, Amazon S3 und zu erhalten. AWS Step Functions
7. Aktivieren Sie unter Integrieren mit AWS Systems Manager OpsCenter das Kontrollkästchen neben AWS Systems Manager OpsCenter OpsItems Für Abhilfemaßnahmen generieren, um Benachrichtigungen anzuzeigen und zu erhalten, wenn bei den ausgewählten Anwendungen Probleme festgestellt werden. Geben Sie einen ARN für das SNS-Thema an, um die Operationen nachzuverfolgen OpsItems, die zur Lösung von so genannten operativen Arbeitsaufgaben ausgeführt werden, die sich auf Ihre AWS Ressourcen beziehen.
8. Sie können optional Tags eingeben, die Ihnen helfen, Ihre Ressourcen zu identifizieren und zu organisieren. CloudWatch Application Insights unterstützt sowohl tagbasierte als auch AWS CloudFormation stapelbasierte Ressourcengruppen, mit Ausnahme von Gruppen. Application Auto Scaling Weitere Informationen finden Sie unter Tag-Editor im Benutzerhandbuch zu AWS Resource Groups und Tags.

9. Um die erkannten Komponenten zu überprüfen, wählen Sie Weiter.
10. Auf der Seite Entdeckte Komponenten überprüfen werden die überwachten Komponenten und ihre Workloads aufgeführt, die von CloudWatch Application Insights automatisch erkannt wurden.
    - Um den Workload-Typ und den Namen zu bearbeiten, wählen Sie Komponente bearbeiten.

> **Note**
>
> Komponenten, die einen erkannten NetWeaver verteilten Workload oder einen Workload NetWeaver mit hoher Verfügbarkeit enthalten, unterstützen nur einen Workload auf einer Komponente.



11. Wählen Sie Weiter aus.
12. Wählen Sie auf der Seite Specify component details (Komponentendetails angeben) die Option Next (Weiter) aus.
13. Überprüfen Sie die Konfiguration der Anwendungsüberwachung und wählen Sie Absenden aus.
14. Die Seite mit den Anwendungsdetails öffnet sich, auf der Sie die Anwendungsübersicht, das Dashboard, die Komponenten und die Workloads sehen können. Sie können auch den Configuration history (Konfigurationsverlauf), Log patterns (Protokollmuster) und jegliche von Ihnen erstellte Tags (Tags) sehen. Nachdem Sie Ihre Anwendung eingereicht haben, stellt CloudWatch Application Insights alle Metriken und Alarme für Ihr NetWeaver SAP-System bereit. Dies kann bis zu einer Stunde dauern.

Verwalten Sie die Überwachung Ihrer NetWeaver SAP-Anwendungsserver

Gehen Sie wie folgt vor, um die Überwachung Ihrer NetWeaver SAP-Anwendungsserver zu verwalten.

So verwalten Sie die Überwachung

1. Öffnen Sie die CloudWatch -Konsole.
2. Wählen Sie im linken Navigationsbereich unter Insights (Einblicke) die Option Application Insights (Anwendungseinblicke) aus.
3. Wählen Sie den Tab List view (Listenansicht).
4. Auf der Seite Application Insights sehen Sie die Anwendungen, die mit Application Insights überwacht werden, samt Überwachungsstatus.
5. Wählen Sie Ihre Anwendung aus.
6. Wählen Sie die Registerkarte Components (Komponenten).
7. Unter Monitored components (Überwachte Komponenten) wählen Sie das Optionsfeld neben dem Komponentennamen aus. Wählen Sie dann Manage monitoring (Überwachung verwalten) aus.
8. Unter Instance logs (Instance-Protokolle) können Sie den vorhandenen Protokollpfad, den Protokollmuster-Satz und den Protokollgruppennamen aktualisieren. Darüber hinaus können Sie bis zu drei zusätzliche Application logs (Anwendungsprotokolle) hinzufügen.
9. Unter Metriken können Sie die NetWeaver SAP-Metriken entsprechend Ihren Anforderungen auswählen. Den Namen der NetWeaver SAP-Metriken wird das Präfix vorangestellt`sap`. Sie können bis zu 40 Metriken pro Komponente hinzufügen.
10. Unter Benutzerdefinierte Alarme können Sie zusätzliche Alarme hinzufügen, die von CloudWatch Application Insights überwacht werden sollen.
11. Überprüfen Sie die Konfiguration der Anwendungsüberwachung und wählen Sie Save (Speichern) aus. Wenn Sie Ihre Konfiguration einreichen, aktualisiert Ihr Konto alle Metriken und Alarme für Ihre NetWeaver SAP-Systeme.

Von CloudWatch Application Insights erkannte NetWeaver SAP-Probleme anzeigen und beheben

In den folgenden Abschnitten finden Sie Schritte, die Sie bei der Lösung häufiger Problemlösungsszenarien unterstützen, die bei der Konfiguration der Überwachung für SAP NetWeaver in Application Insights auftreten.

Themen zur Fehlerbehebung

- Probleme mit der NetWeaver SAP-Datenbankkonnektivität
- Probleme mit NetWeaver der Verfügbarkeit von SAP-Anwendungen

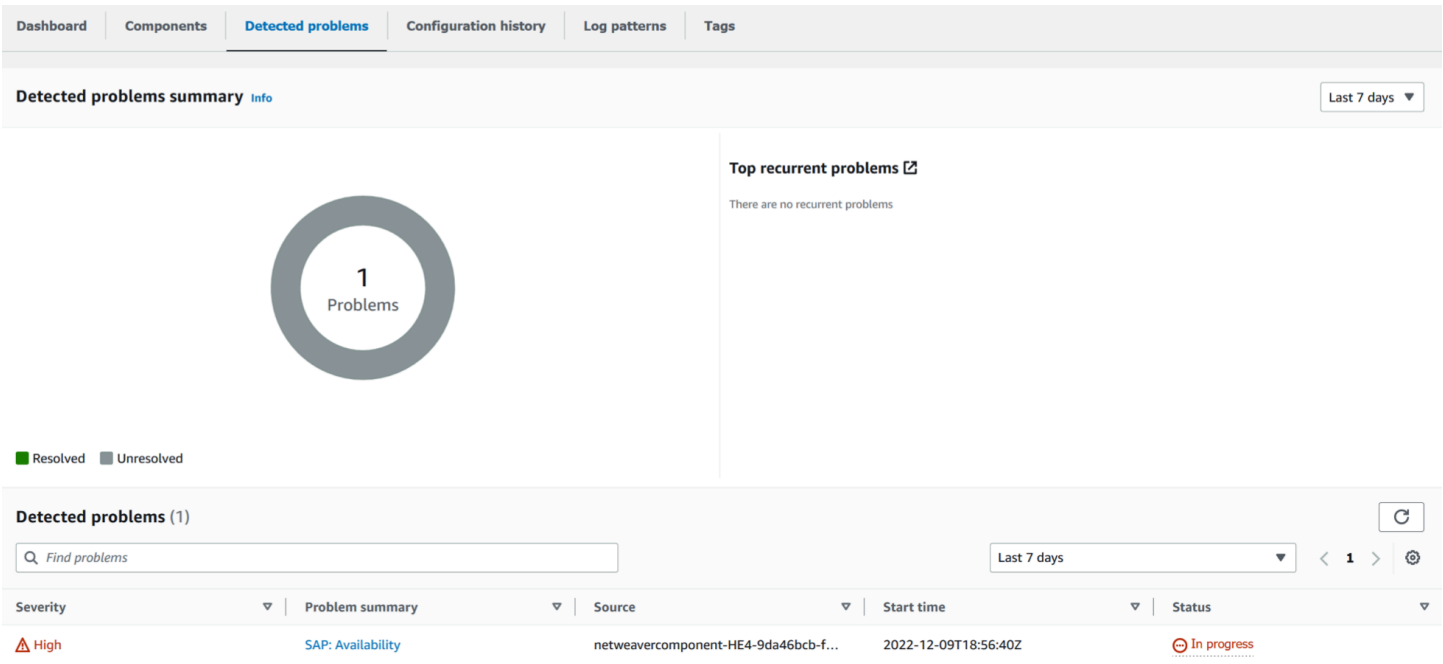
Probleme mit der NetWeaver SAP-Datenbankkonnektivität

Beschreibung

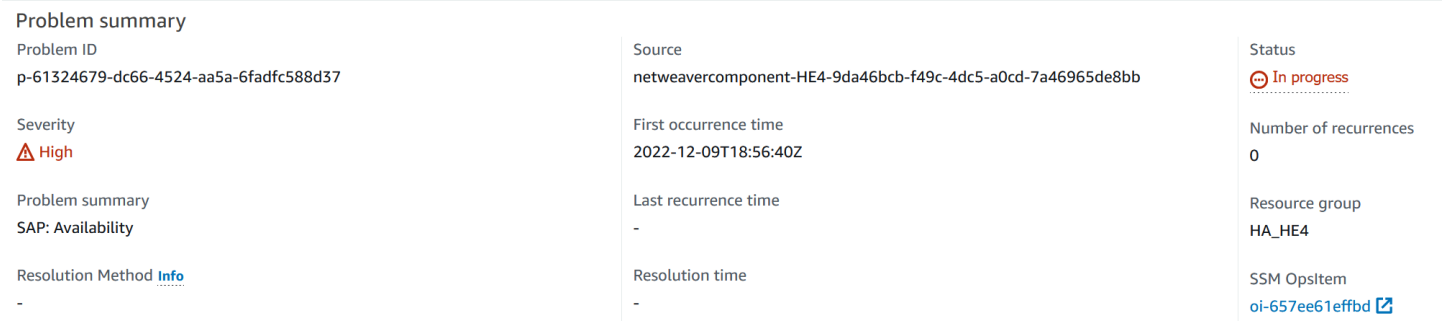
In Ihrer NetWeaver SAP-Anwendung treten Probleme mit der Datenbankkonnektivität auf.

Ursache

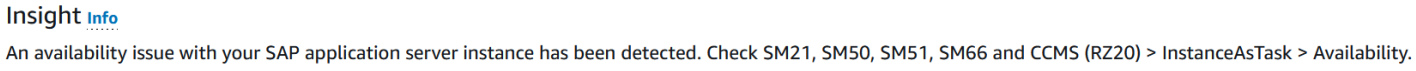
Sie können das Verbindungsproblem identifizieren, indem Sie zur CloudWatch Application Insights-Konsole gehen und das SAP NetWeaver Application Insights-Problem-Dashboard überprüfen. Wählen Sie den Link unter Problem summary (Problemzusammenfassung) aus, um das spezifische Problem zu sehen.



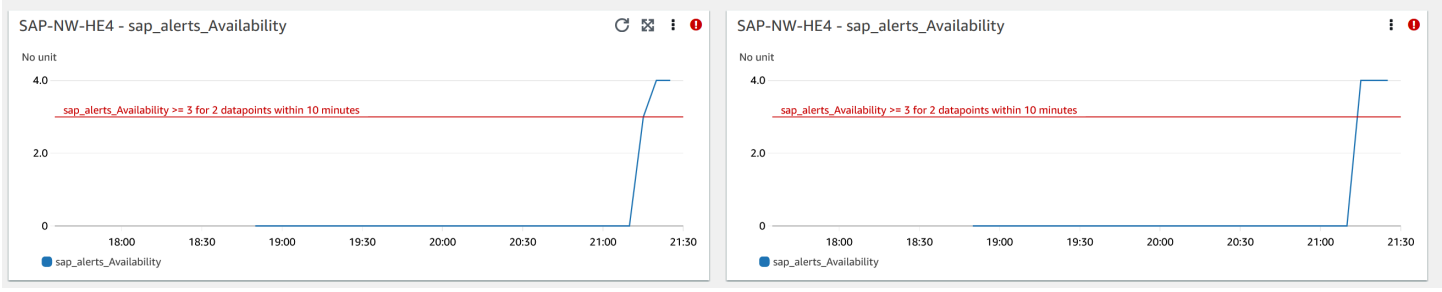
Im folgenden Beispiel ist unter Problem summary (Problemzusammenfassung) SAP: Verfügbarkeit das Problem.



Unmittelbar nach der Problem summary (Problemzusammenfassung) bietet der Abschnitt Insight (Einblicke) mehr Kontext zum Fehler und weitere Informationen zu den Ursachen des Problems.

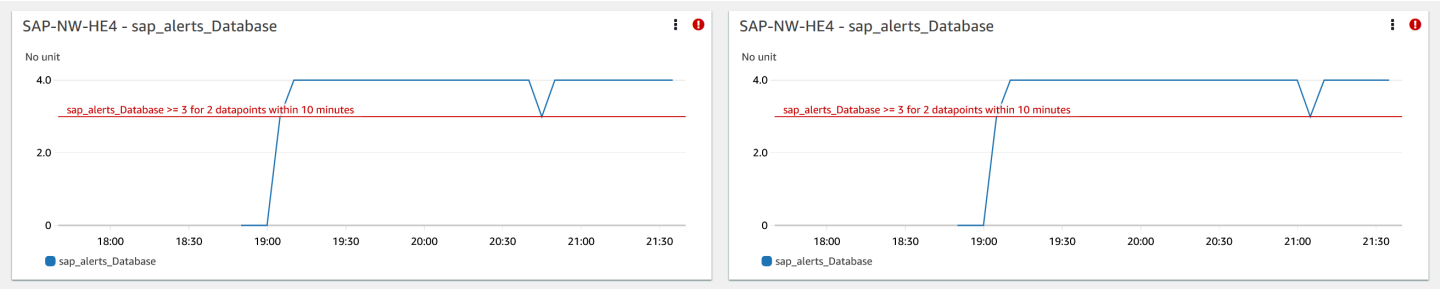


Auf demselben Problem-Dashboard können Sie sich die zugehörigen Protokolle und Metriken ansehen, die die Problemerkennung gruppiert hat, um die Ursache des Fehlers zu ermitteln. Die `sap_alerts_Availability` Metrik verfolgt die Verfügbarkeit des NetWeaver SAP-Systems im Laufe der Zeit. Mithilfe der historischen Nachverfolgung können Sie korrelieren, wann die Metrik einen Fehlerstatus ausgelöst hat oder die Alarmschwelle überschritten wurde. Im folgenden Beispiel liegt ein Verfügbarkeitsproblem mit dem NetWeaver SAP-System vor. Das Beispiel zeigt zwei Alarme, da es zwei SAP-Anwendungsserver-Instances gibt und für jede Instance ein Alarm erstellt wurde.

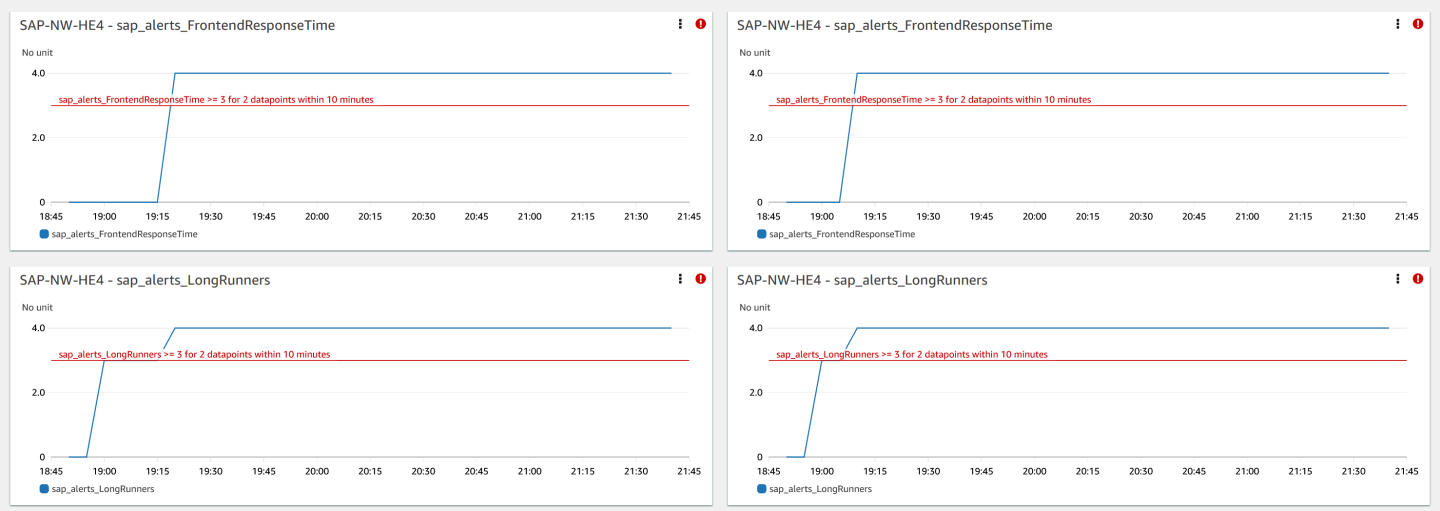


Um weitere Informationen zu den einzelnen Alarmen anzuzeigen, fahren Sie mit der Maus über den `sap_alerts_Availability`-Metriknamen.

Im folgenden Beispiel zeigt die `sap_alerts_Database`-Metrik, dass die Datenbankschicht ein Problem oder einen Fehler aufweist. Dieser Alarm weist darauf hin, dass SAP Probleme NetWeaver hatte, eine Verbindung zu seiner Datenbank herzustellen oder mit ihr zu kommunizieren.



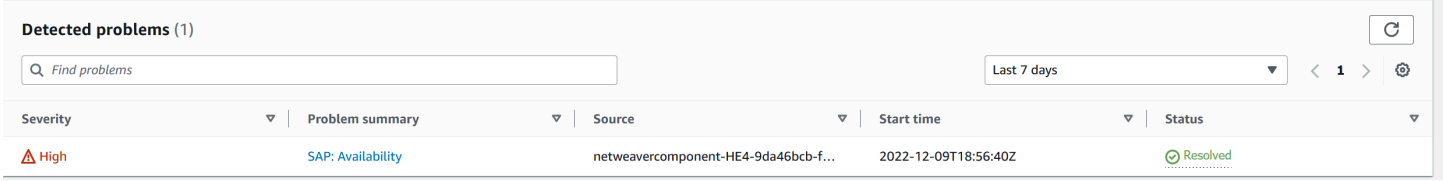
Da die Datenbank eine wichtige Ressource für SAP ist NetWeaver, erhalten Sie möglicherweise viele entsprechende Alarme, wenn in der Datenbank ein Problem oder ein Ausfall auftritt. Im folgenden Beispiel werden die `sap_alerts_FrontendResponseTime`- und `sap_alerts_LongRunners`-Metriken initiiert, da die Datenbank nicht verfügbar ist.



Auflösung

Application Insights überwacht das erkannte Problem stündlich. Wenn Ihre SAP-Protokolldateien keine neuen zugehörigen NetWeaver Protokolleinträge enthalten, werden die älteren Protokolleinträge als behoben behandelt. Sie müssen alle Fehlerbedingungen im Zusammenhang mit den CloudWatch Alarmen beheben. Nachdem die Fehlerbedingungen behoben wurden, wird der Alarm behoben, wenn die Alarme und Protokolle wiederhergestellt sind. Wenn alle CloudWatch Protokollfehler und Alarme behoben sind, erkennt Application Insights keine Fehler mehr und das Problem wird innerhalb einer Stunde automatisch behoben. Wir empfehlen Ihnen, alle Protokollfehler und Alarme zu beheben, damit Sie die neuesten Probleme im Problem-Dashboard angezeigt bekommen.

Im folgenden Beispiel ist das Problem der SAP-Verfügbarkeit gelöst.



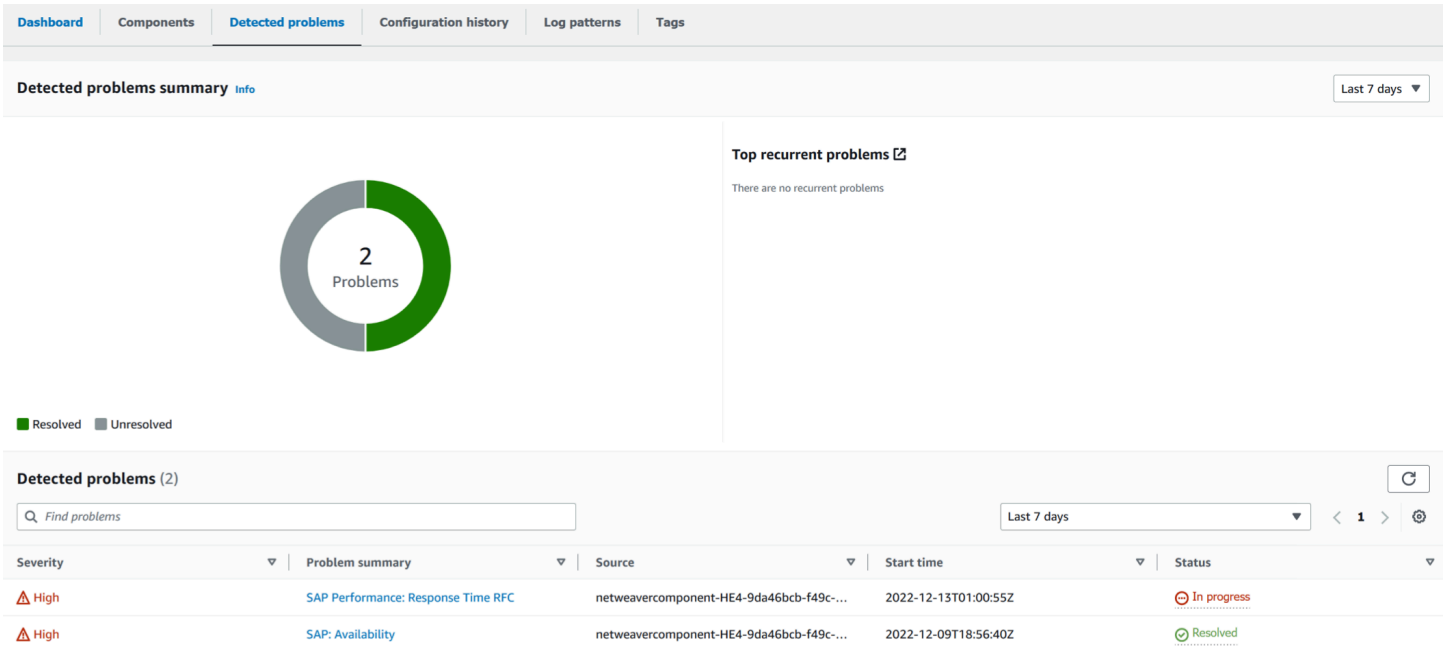
Probleme mit NetWeaver der Verfügbarkeit von SAP-Anwendungen

Beschreibung

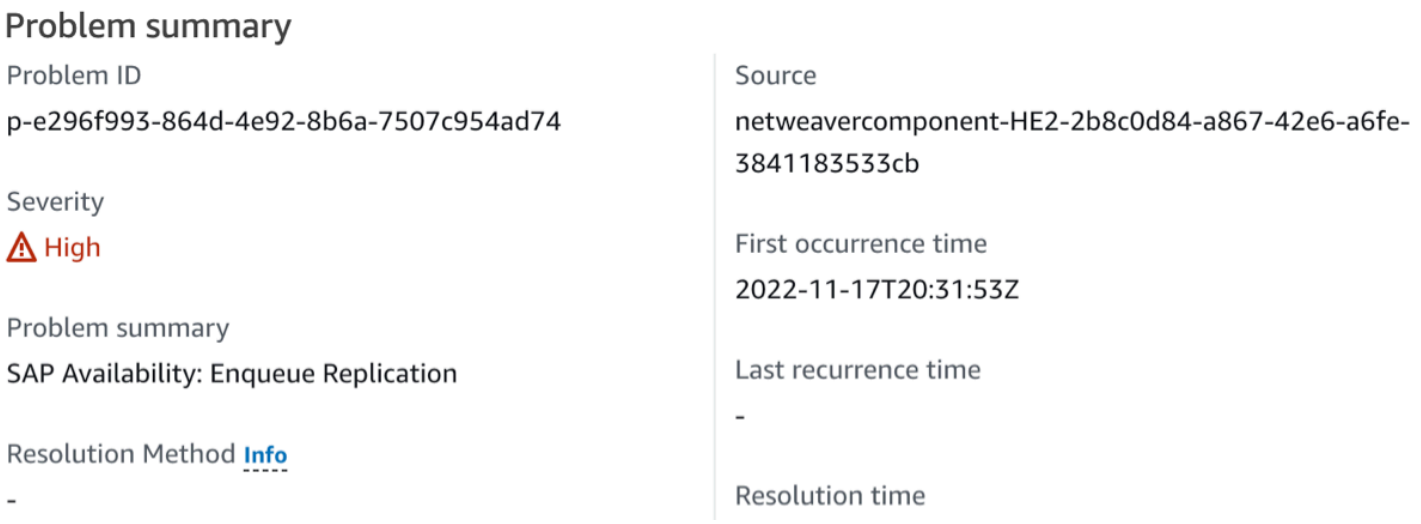
Ihre SAP NetWeaver High Availability Enqueue-Replikation funktioniert nicht mehr.

Ursache

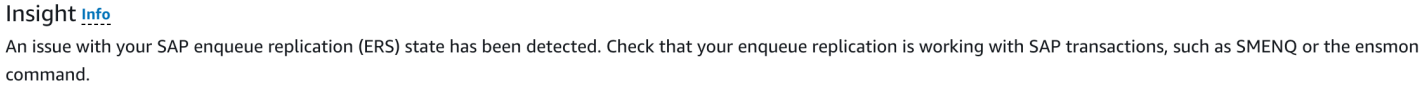
Sie können das Verbindungsproblem identifizieren, indem Sie zur CloudWatch Application Insights-Konsole gehen und das SAP NetWeaver Application Insights-Problem-Dashboard überprüfen. Wählen Sie den Link unter Problem summary (Problemzusammenfassung) aus, um das spezifische Problem zu sehen.



Im folgenden Beispiel ist unter Problem summary (Problemzusammenfassung) die Hochverfügbarkeits-Enqueue-Replikation das Problem.

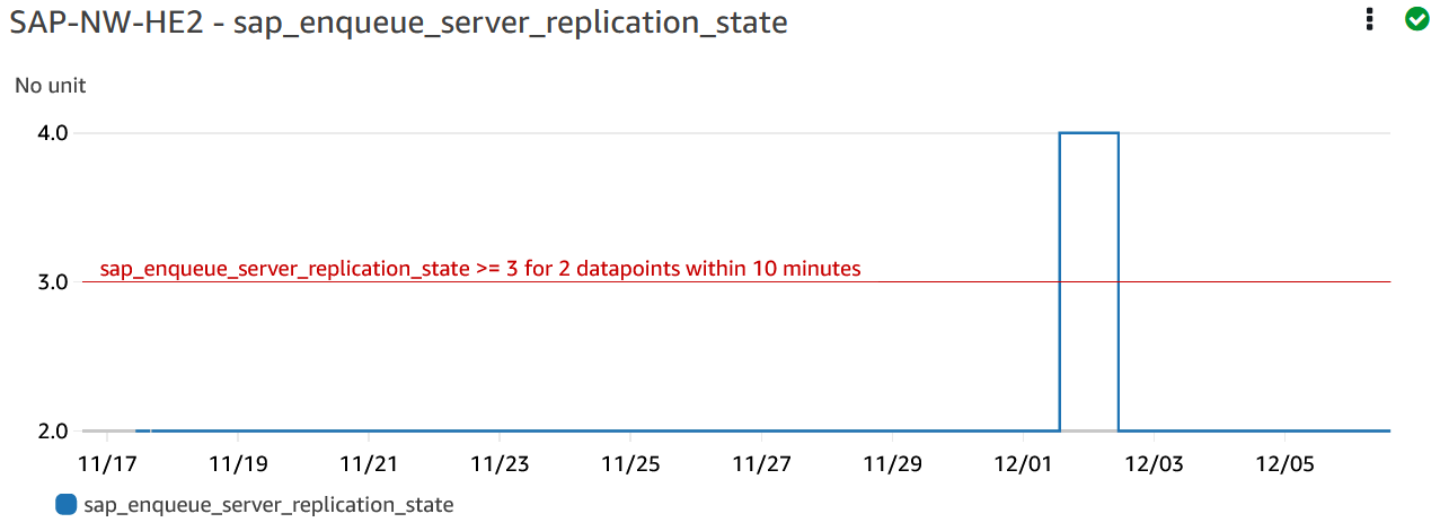


Unmittelbar nach der Problem summary (Problemzusammenfassung) bietet der Abschnitt Insight (Einblicke) mehr Kontext zum Fehler und weitere Informationen zu den Ursachen des Problems.

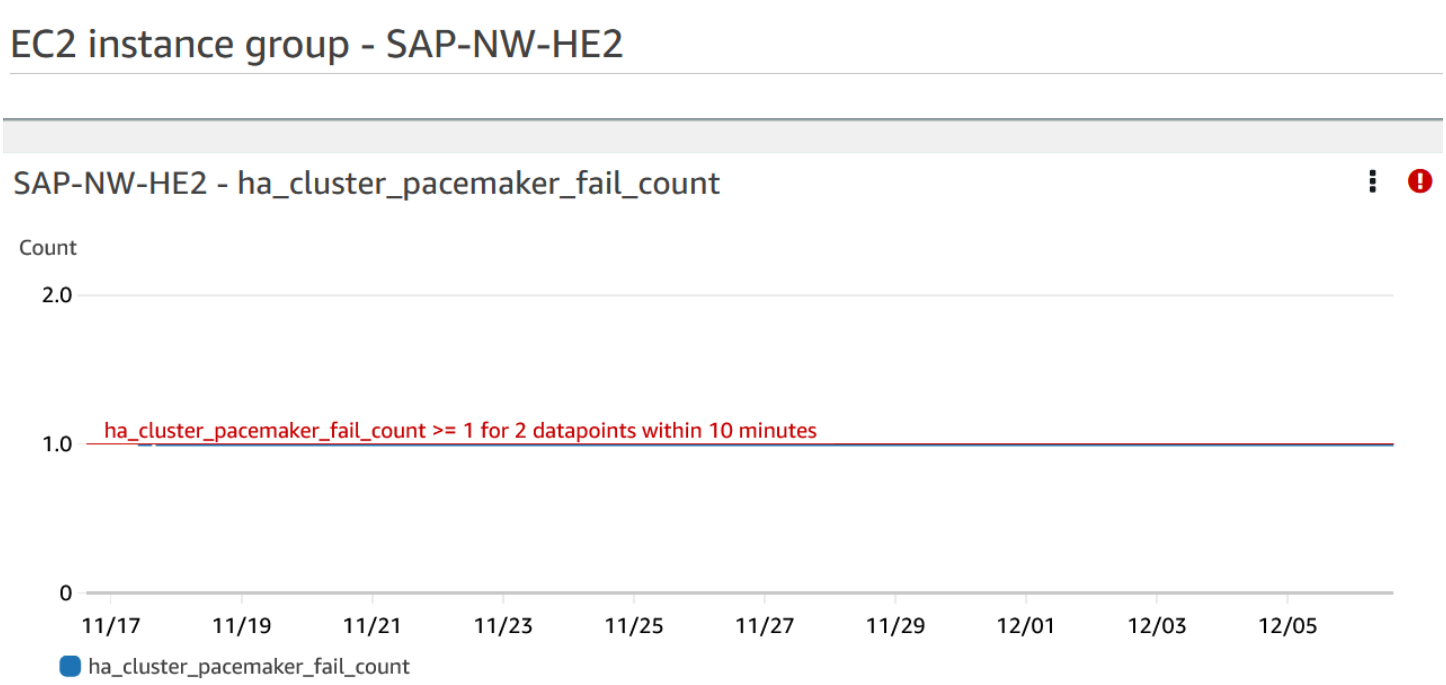


Das folgende Beispiel zeigt das Problem-Dashboard, in dem Sie Protokolle und Metriken anzeigen, die gruppiert sind, um Ihnen zu helfen, die Fehlerursachen zu identifizieren. Die `sap_enqueue_server_replication_state`-Metrik verfolgt den Wert im Zeitverlauf. Mithilfe der

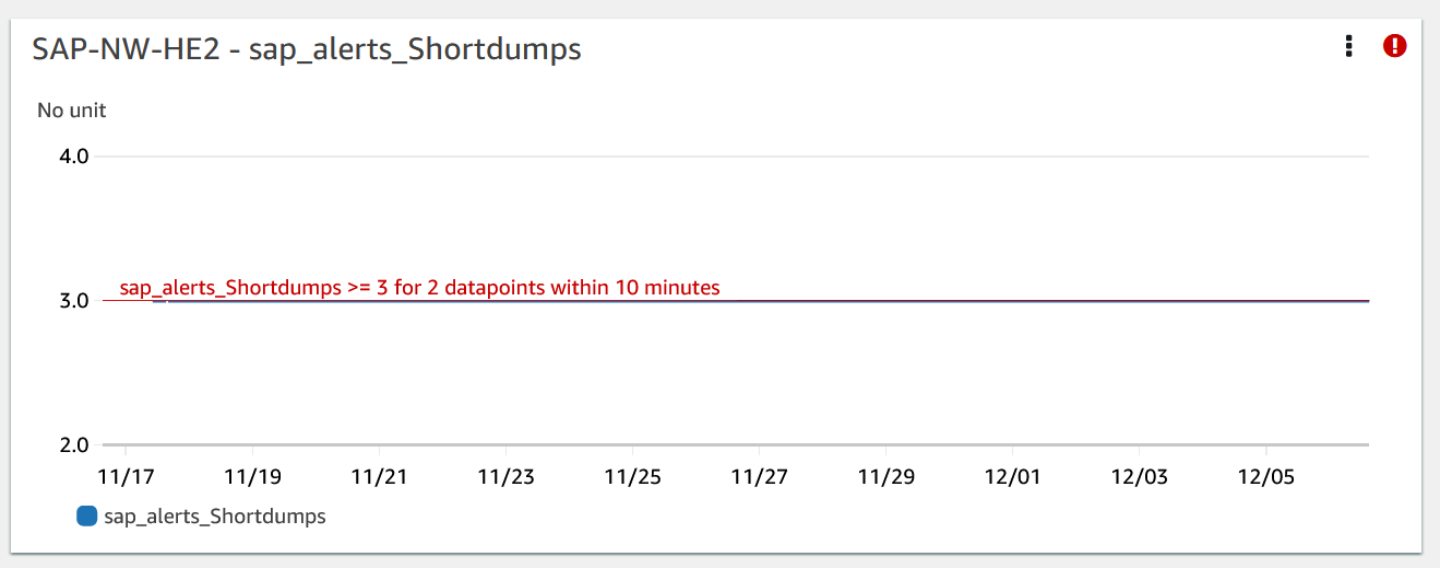
historischen Nachverfolgung können Sie korrelieren, wann die Metrik einen Fehlerstatus ausgelöst hat oder die Alarmschwelle überschritten wurde.



Im folgenden Beispiel zeigt die `ha_cluster_pacemaker_fail_count`-Metrik, dass im Pacemaker-Cluster mit hoher Verfügbarkeit ein Ressourcenausfall aufgetreten ist. Die spezifischen Pacemaker-Ressourcen, bei denen die Anzahl der Fehler größer oder gleich eins war, werden im Komponenten-Dashboard identifiziert.

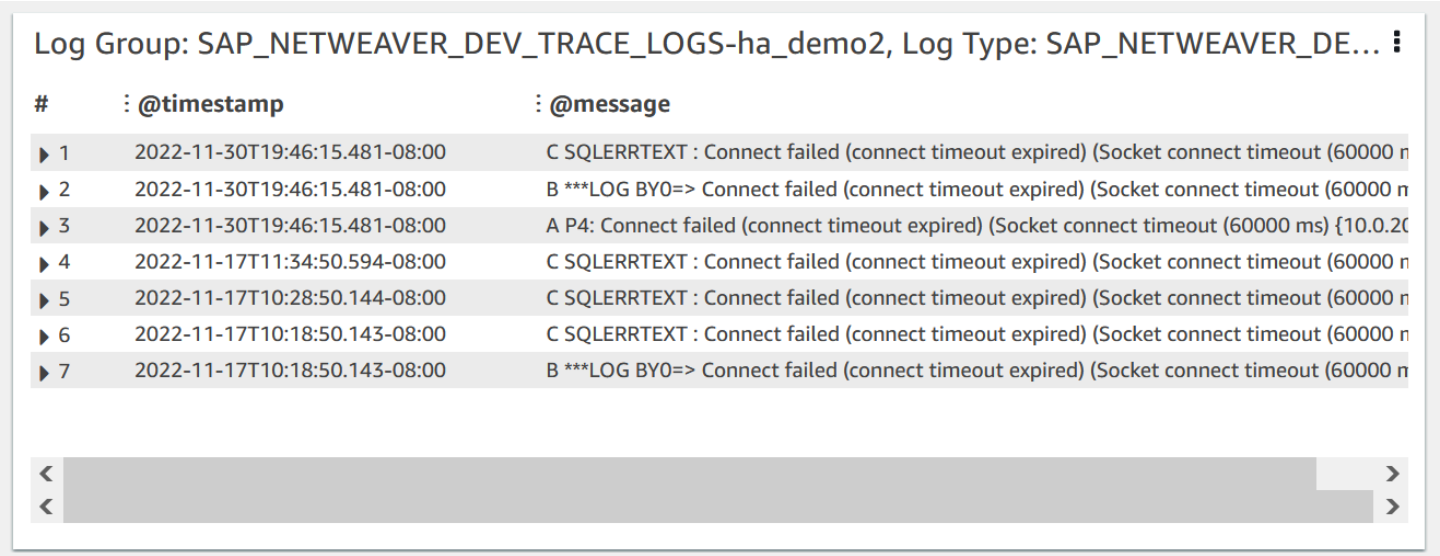


Das folgende Beispiel zeigt die `sap_alerts_Shortdumps`-Metrik, die angibt, dass die Leistung der SAP-Anwendung reduziert wurde, als das Problem erkannt wurde.



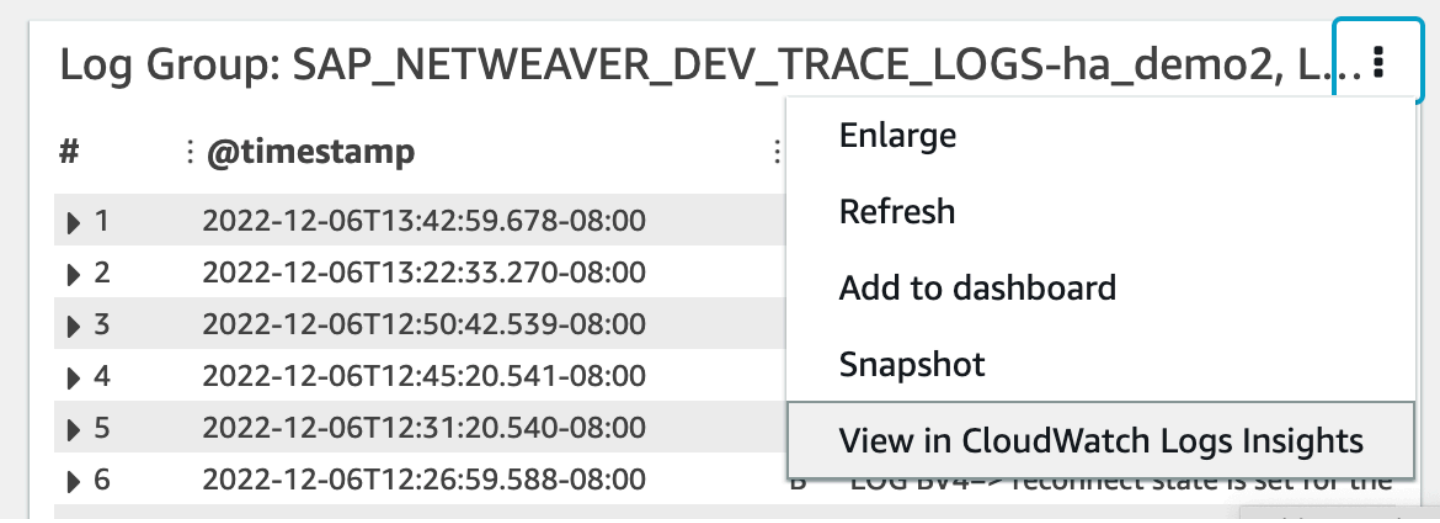
Logs (Protokolle)

Die Protokolleinträge sind hilfreich, um ein besseres Verständnis der Probleme zu erhalten, die auf NetWeaver SAP-Ebene aufgetreten sind, als das Problem erkannt wurde. Das Protokollgruppen-Widget im Problem-Dashboard zeigt die spezifische Zeit des Problems.



#	@timestamp	@message
1	2022-11-30T19:46:15.481-08:00	C SQLERRTEXT : Connect failed (connect timeout expired) (Socket connect timeout (60000 m
2	2022-11-30T19:46:15.481-08:00	B ***LOG BY0=> Connect failed (connect timeout expired) (Socket connect timeout (60000 m
3	2022-11-30T19:46:15.481-08:00	A P4: Connect failed (connect timeout expired) (Socket connect timeout (60000 ms) {10.0.2(
4	2022-11-17T11:34:50.594-08:00	C SQLERRTEXT : Connect failed (connect timeout expired) (Socket connect timeout (60000 m
5	2022-11-17T10:28:50.144-08:00	C SQLERRTEXT : Connect failed (connect timeout expired) (Socket connect timeout (60000 m
6	2022-11-17T10:18:50.143-08:00	C SQLERRTEXT : Connect failed (connect timeout expired) (Socket connect timeout (60000 m
7	2022-11-17T10:18:50.143-08:00	B ***LOG BY0=> Connect failed (connect timeout expired) (Socket connect timeout (60000 m

Um detaillierte Informationen zu den Protokollen anzuzeigen, wählen Sie die drei vertikalen Punkte in der oberen rechten Ecke aus und wählen Sie In CloudWatch Logs Insights anzeigen aus.

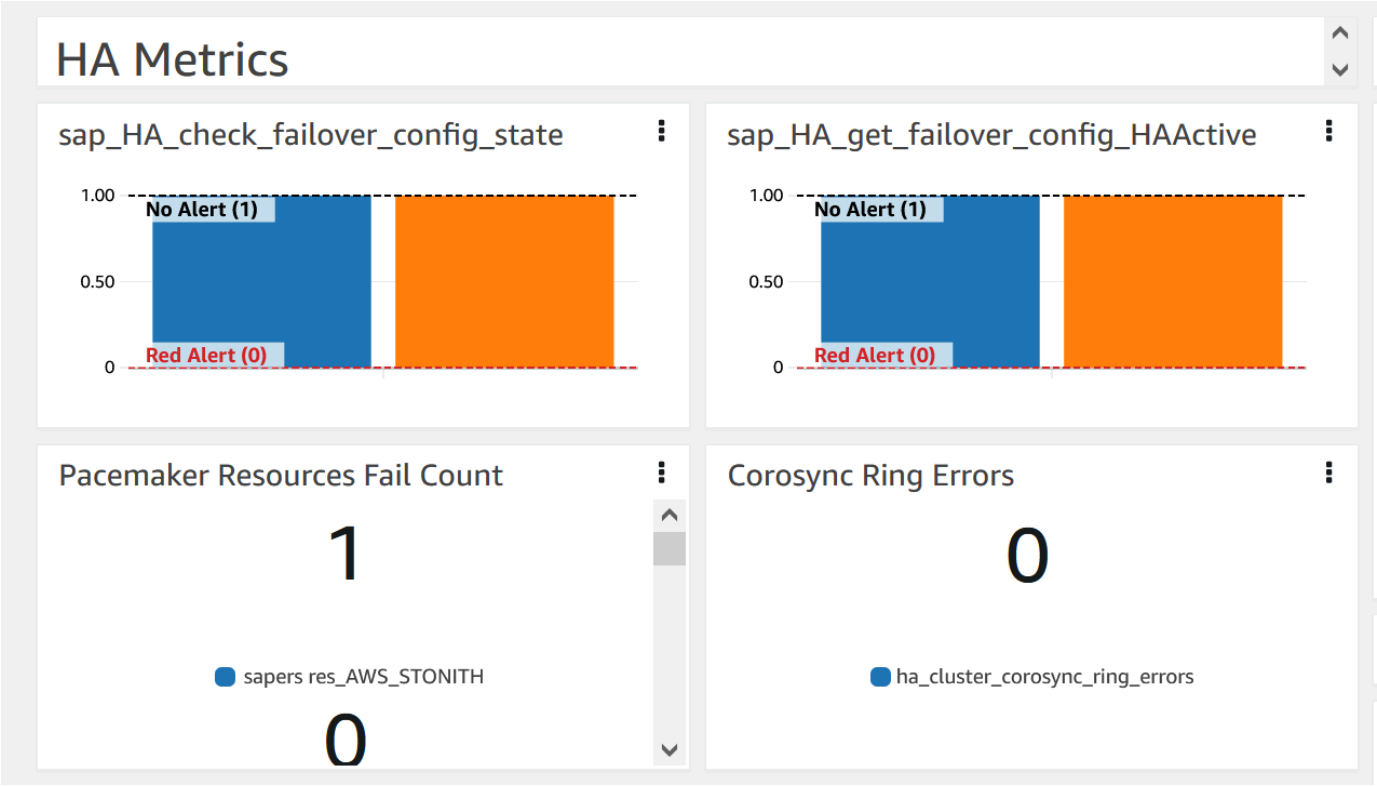


Folgen Sie diesen Schritten, um weitere Informationen zu den Metriken und Alarmen zu erhalten, die im Problem-Dashboard angezeigt werden.

So erhalten Sie weitere Informationen zu Metriken und Alarmen

1. Öffnen Sie die CloudWatch -Konsole.
2. Wählen Sie im linken Navigationsbereich unter Insights (Einblicke) die Option Application Insights (Anwendungseinblicke) aus. Wählen Sie dann die Registerkarte List view (Listenansicht) und wählen Sie Ihre Anwendung aus.
3. Wählen Sie die Registerkarte Components (Komponenten) aus. Wählen Sie dann die NetWeaver SAP-Komponente aus, zu der Sie weitere Informationen erhalten möchten.

   Das folgende Beispiel zeigt den Abschnitt HA Metrics (HA-Metriken) mit der `ha_cluster_pacemaker_fail_count`-Metrik, die im Problem-Dashboard angezeigt wurde.

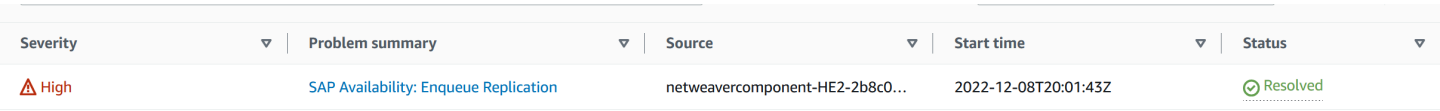


Auflösung

Application Insights überwacht das erkannte Problem stündlich. Wenn Ihre SAP-Protokolldateien keine neuen zugehörigen NetWeaver Protokolleinträge enthalten, werden die älteren Protokolleinträge als behoben behandelt. Sie müssen alle mit diesem Problem verbundenen Fehlerbedingungen beheben.

Für den `sap_alerts_Shortdumps` Alarm müssen Sie den Alert im NetWeaver SAP-System auflösen, indem Sie den Transaktionscode verwenden`RZ20 # R3Abap # Shortdumps`, um zum CCMS-Alert zu navigieren. Weitere Informationen über CCMS-Warungen finden Sie auf der SAP-Website. Beheben Sie alle CCMS-Warnungen im Shortdumps-Baum. Nachdem alle Alerts im NetWeaver SAP-System behoben CloudWatch wurden, meldet die Metrik nicht mehr den Alarmstatus.

Wenn alle CloudWatch Protokollfehler und Alarme behoben sind, erkennt Application Insights keine Fehler mehr und das Problem wird innerhalb einer Stunde automatisch behoben. Wir empfehlen Ihnen, alle Protokollfehler und Alarme zu beheben, damit Sie die neuesten Probleme im Problem-Dashboard angezeigt bekommen. Im folgenden Beispiel wird das Enqueue-Replikations-Problem von SAP Netweaver High Availability gelöst.



Severity	Problem summary	Source	Start time	Status
High	SAP Availability: Enqueue Replication	netweavercomponent-HE2-2b8c0...	2022-12-08T20:01:43Z	Resolved

Fehlerbehebung bei Application Insights für SAP NetWeaver

Dieser Abschnitt enthält Schritte, mit denen Sie häufige Fehler beheben können, die vom Application-Insights-Dashboard zurückgegeben werden.

Es können nicht mehr als 60 Überwachungsmetriken hinzugefügt werden

Zurückgegebener Fehler: `Component cannot have more than 60 monitored metrics.` (Komponente kann nicht mehr als 60 überwachte Metriken haben)

Ursache: `The current metric limit is 60 monitor metrics per component.` (Das aktuelle Limit sind 60 überwachte Metriken pro Komponente)

Lösung: Entfernen Sie Metriken, die nicht erforderlich sind, um das Limit einzuhalten.

SAP-Metriken werden nach dem Einbindungs-Prozess nicht im Dashboard angezeigt

Ursache: Das Komponenten-Dashboard verwendet einen metrischen Zeitraum von fünf Minuten, um die Datenpunkte zu aggregieren.

Lösung: Alle Metriken sollten nach fünf Minuten im Dashboard angezeigt werden.

SAP-Metriken und Alarme werden nicht im Dashboard angezeigt

Gehen Sie wie folgt vor, um zu ermitteln, warum SAP-Metriken und Alarme nach dem Einbindungs-Prozess nicht im Dashboard angezeigt werden.

So identifizieren Sie das Problem anhand von Metriken und Alarmen

1. Öffnen Sie die CloudWatch -Konsole.
2. Wählen Sie im linken Navigationsbereich unter Insights (Einblicke) die Option Application Insights (Anwendungseinblicke) aus. Wählen Sie dann die Registerkarte List view (Listenansicht) und wählen Sie Ihre Anwendung aus.
3. Wählen Sie die Registerkarte Configuration history (Konfigurations-Historie) aus.
4. Wenn Sie fehlende Metrikdatenpunkte sehen, suchen Sie nach Fehlern im Zusammenhang mit `prometheus-sap_host_exporter`.

5. Wenn Sie im vorherigen Schritt keinen Fehler finden, stellen Sie eine Verbindung zu Ihrer Linux-Instance her. Stellen Sie für Hochverfügbarkeitsbereitstellungen eine Verbindung zur primären Amazon-EC2-Cluster-Instance her.
6. Vergewissern Sie sich in Ihrer Instance, dass der Exporter läuft, indem Sie den folgenden Befehl verwenden. Der Standard-Port ist `9680`. Wenn Sie einen anderen Port verwenden, ersetzen Sie `9680` durch den Port, den Sie verwenden.

   ```
   curl localhost:9680/metrics
   ```

   Wenn keine Daten zurückgegeben werden, konnte der Exporter nicht gestartet werden.
7. Führen Sie den folgenden Befehl aus, um die richtige Benennungskonvention für `WORKLOAD_SHORT_NAME` die nächsten beiden Schritte zu finden.

   > **Note**
   >
   > Application Insights fügt dem Dienstnamen je nach ausgeführtem Workload ein Suffix,, hinzu. `WORKLOAD_SHORT_NAME` Die Kurznamen für NetWeaver verteilte Bereitstellungen, Standardbereitstellungen und Bereitstellungen mit hoher Verfügbarkeit lauten `SAP_NWDSAP_NWS`, und. `SAP_NWH`

   ```
   sudo systemctl | grep exporter
   ```
8. Führen Sie den folgenden Befehl aus, um nach Fehlern in den Exportdienstprotokollen zu suchen:

   ```
   sudo journalctl -e --unit=prometheus-sap_host_exporter_WORKLOAD_SHORT_NAME.service
   ```
9. Führen Sie den folgenden Befehl aus, um nach Fehlern in den Dienstprotokollen des Export-Managers zu suchen:

   ```
   sudo journalctl -e --unit=prometheus-
   sap_host_exporter_manager_WORKLOAD_SHORT_NAME.service
   ```

> **Note**
>
> Dieser Service sollte jederzeit verfügbar sein.

Wenn dieser Befehl keinen Fehler zurückgibt, fahren Sie mit dem nächsten Schritt fort.

10. Führen Sie den folgenden Befehl aus, um den Exporter manuell zu starten. Überprüfen Sie danach die Ausgabe des Exporters.

```
sudo /opt/aws/sap_host_exporter/sap_host_exporter
```

Sie können den Exportervorgang beenden, nachdem Sie nach Fehlern gesucht haben.

Fehlerursache: Für dieses Problem gibt es verschiedene mögliche Ursachen. Eine häufige Ursache ist, dass der Exporter keine Verbindung zu einer der Anwendungsserver-Instances herstellen kann.

Resolution (Auflösung)

Gehen Sie wie folgt vor, um den Exporter mit den Anwendungsserver-Instances zu verbinden. Sie überprüfen, ob die SAP-Anwendungsinstance läuft, und verwenden SAPControl, um sich mit der Instance zu verbinden.

So verbinden Sie den Exporter mit den Anwendungsserver-Instances

1. Führen Sie in Ihrer Amazon-EC2-Instance den folgenden Befehl aus, um zu überprüfen, ob die SAP-Anwendung ausgeführt wird.

```
sapcontrol -nr <App_InstNo> -function GetProcessList
```

2. Sie müssen eine funktionierende SAPControl-Verbindung aufbauen. Wenn die SAPControl-Verbindung nicht funktioniert, suchen Sie die Ursache des Problems in der entsprechenden SAP-Anwendungs-Instance.
3. Führen Sie den folgenden Befehl aus, um den Exporter manuell zu starten, nachdem Sie das SAPControl-Verbindungsproblem behoben haben:

```
sudo systemctl start prometheus-sap_host_exporter.service
```

4. Wenn Sie das SAPControl-Verbindungsproblem nicht beheben können, verwenden Sie das folgende Verfahren als Übergangslösung.

   a. Öffnen Sie die AWS Systems Manager -Konsole.

   b. Wählen Sie im linken Navigationsbereich State Manager (Statusverwaltung).

   c. Suchen Sie unter Assoziationen nach der Assoziation des NetWeaver SAP-Systems.

   ```
   Association Name: Equal: AWS-ApplicationInsights-
   SSMSAPHostExporterAssociationForCUSTOMSAPNW<SID>-1
   ```

   d. Wählen Sie die Association id (Assoziations-ID) aus.

   e. Wählen Sie die Registerkarte Parameter (Parameter) und entfernen Sie die Anwendungsservernummer aus additionalArguments (zusätzliche Argumente).

   f. Wählen Sie Apply association now (Zuordnung jetzt anwenden).

   > **Note**
   >
   > Dies ist eine Übergangslösung. Wenn die Überwachungskonfigurationen der Komponente aktualisiert werden, wird die Instance wieder hinzugefügt.

Von Amazon CloudWatch Application Insights erkannte Probleme anzeigen und beheben

Die Themen in diesem Abschnitt enthalten detaillierte Informationen zu den erkannten Problemen und Erkenntnissen, die von Application Insights angezeigt werden. Es werden auch Lösungsvorschläge für erkannte Probleme mit Ihrem Konto bzw. Ihrer Konfiguration angeboten.

Themen zur Fehlerbehebung

- CloudWatch Übersicht über die Konsole
- Übersichtsseite zu Problemen von Application Insights
- CloudWatch Fehler bei der Zusammenführung von Agenten
- Alarme werden nicht erstellt
- Feedback
- Konfigurationsfehler

CloudWatch Übersicht über die Konsole

Eine Übersicht der Probleme, die sich auf Ihre überwachten Anwendungen auswirken, finden Sie im Bereich CloudWatch Application Insights auf der Übersichtsseite der CloudWatch Konsole. Weitere Informationen finden Sie unter Erste Schritte mit Amazon CloudWatch Application Insights.

Im Übersichtsbereich von CloudWatch Application Insights wird Folgendes angezeigt:

- Den Schweregrad der festgestellten Probleme (hoch/mittel/niedrig)
- Eine kurze Zusammenfassung des Problems.
- Die Problemquelle
- Die Zeit, zu der das Problem begann.
- Der Lösungsstatus des Problems.
- Die betroffene Ressourcengruppe

Um ein bestimmtes Problem detailliert zu untersuchen, wählen Sie unter Problem Summary (Problemübersicht) die Beschreibung des Problems aus. Ein detailliertes Dashboard zeigt Einblicke in das Problem und die damit verbundenen Metrikanomalien und Ausschnitte von Protokollfehlern. Hier können Sie Feedback zur Relevanz der Erkenntnisse geben, indem Sie auswählen, ob sie nützlich sind.

Wenn eine neue, nicht konfigurierte Ressource erkannt wird, führt sie die Problembeschreibung zum Assistenten Edit configuration (Konfiguration bearbeiten), um Ihre neue Ressource zu konfigurieren. Sie können Ihre Ressourcengruppenkonfiguration anzeigen oder bearbeiten, indem Sie View/edit configuration (Konfiguration anzeigen/bearbeiten) in der oberen rechten Ecke des detaillierten Dashboards auswählen.

Um zur Übersicht zurückzukehren, wählen Sie Zurück zur Übersicht aus. Diese Option befindet sich neben der detaillierten Überschrift des CloudWatch Application Insights-Dashboards.

Übersichtsseite zu Problemen von Application Insights

Übersichtsseite zu Problemen von Application Insights

CloudWatch Application Insights bietet auf der Seite mit der Problemzusammenfassung die folgenden Informationen zu erkannten Problemen:

- Eine kurze Zusammenfassung des Problems.
- Die Startzeit und das Datum des Problems.

- Der Schweregrad des Problems: Hoch/Mittel/Niedrig.
- Der Status des erkannten Problems: in Bearbeitung/gelöst.
- Einblicke: Automatisch generierte Erkenntnisse über das erkannte Problem und die mögliche Hauptursache.
- Feedback zu Erkenntnissen: Feedback, das Sie zur Nützlichkeit der durch CloudWatch Application Insights generierten Erkenntnisse gegeben haben
- Verwandte Beobachtungen: Eine detaillierte Übersicht über die Metrikanomalien und Fehlerausschnitte relevanter Protokolle im Zusammenhang mit dem Problem über verschiedene Anwendungskomponenten hinweg.

CloudWatch Fehler bei der Zusammenführung von Agenten

CloudWatch Application Insights installiert und konfiguriert den CloudWatch Agenten auf Kundeninstanzen. Dazu gehört die Erstellung einer CloudWatch Agentenkonfigurationsdatei mit Konfigurationen für Metriken oder Protokolle. Ein Zusammenführungskonflikt kann auftreten, wenn für die Instanz eines Kunden bereits eine CloudWatch Agentenkonfigurationsdatei mit unterschiedlichen Konfigurationen für dieselben Metriken oder Protokolle definiert wurde. Um den Zusammenführungskonflikt zu lösen, gehen Sie wie folgt vor:

1. Identifizieren Sie die CloudWatch Agenten-Konfigurationsdateien auf Ihrem System. Weitere Informationen zu den Speicherorten der Dateien finden Sie unter CloudWatch Agentendateien und Speicherorte.
2. Entfernen Sie die Ressourcenkonfigurationen, die Sie in Application Insights verwenden möchten, aus der vorhandenen CloudWatch Agentenkonfigurationsdatei. Wenn Sie nur Application Insights-Konfigurationen verwenden möchten, löschen Sie die vorhandenen CloudWatch Agentenkonfigurationsdateien.

Alarme werden nicht erstellt

Bei einigen Metriken prognostiziert Application Insights die Alarmschwelle auf der Grundlage früherer Datenpunkte für die Metrik. Um diese Vorhersage zu ermöglichen, müssen die folgenden Kriterien erfüllt sein.

- Aktuelle Datenpunkte – Es müssen mindestens 100 Datenpunkte aus den letzten 24 Stunden vorhanden sein. Die Datenpunkte müssen nicht fortlaufend sein, sondern können über den 24-Stunden-Zeitraum verteilt sein.

- Historische Daten – Es müssen mindestens 100 Datenpunkte vorhanden sein, die sich über den Zeitraum von 15 Tagen vor dem aktuellen Datum bis 1 Tag vor dem aktuellen Datum erstrecken. Die Datenpunkte müssen nicht fortlaufend sein, sondern können über den 15-Tage-Zeitraum verteilt sein.

> **Note**
>
> Bei einigen Metriken verzögert Application Insights die Erstellung von Alarmen, bis die vorhergehenden Bedingungen erfüllt sind. In diesem Fall erhalten Sie ein Konfigurationsprotokoll, das besagt, dass für die Metrik nicht genügend Datenpunkte vorhanden sind, um die Alarmschwelle festzulegen.

Feedback

Feedback

Sie können Feedback zu den automatisch generierten Erkenntnissen über erkannte Probleme geben, indem Sie sie als nützlich oder nicht nützlich einstufen. Ihr Feedback zu den Erkenntnissen sowie Ihre Anwendungsdiagnose (Metrikanomalien und Protokollausnahmen) werden genutzt, um die zukünftige Erkennung ähnlicher Probleme zu verbessern.

Konfigurationsfehler

CloudWatch Application Insights verwendet Ihre Konfiguration, um Überwachungstelemetrien für die Komponenten zu erstellen. Wenn Application Insights ein Problem mit Ihrem Konto oder Ihrer Konfiguration erkennt, werden auf der Problemübersichtsseite im Feld Remarks (Bemerkungen) Informationen darüber angezeigt, wie Sie das Konfigurationsproblem für Ihre Anwendung lösen können.

Die folgende Tabelle zeigt Lösungsvorschläge für bestimmte Bemerkungen.

Anmerkungen	Empfohlene Auflösung	Weitere Hinweise
Das Kontingent für CloudFormation wurde bereits erreicht.	Application Insights erstellt für jede Anwendung einen CloudFormation Stack, um die CloudWatch Agentenin	–

Anmerkungen	Empfohlene Auflösung	Weitere Hinweise
	stallation und -konfiguration für alle Anwendungskomponenten zu verwalten. Standardmäßig kann jedes AWS Konto 2000 Stapel haben. Weitere Informationen finden Sie unter AWS CloudFormation -Limits. Um dieses Problem zu beheben, erhöhen Sie das Limit für CloudFormation Stacks.	
Keine SSM-Instance-Rolle für die folgenden Instances.	Damit Application Insights CloudWatch Agenten auf Anwendungsinstanzen installieren und konfigurieren kann, müssen AmazonSSM ManagedInstanceCore und CloudWatchAgentServerPolicy Richtlinien an die Instance-Rolle angehängt werden.	Application Insights ruft die DescribeInstanceInformation SSM-API auf, um die Liste der Instances mit SSM-Berechtigungen abzurufen . Nachdem die Rolle der Instanz zugewiesen wurde, dauert es einige Zeit, bis SSM die Instanz in das Ergebnis aufgenommen hat. DescribeInstanceInformation Bis SSM die Instance in das Ergebnis aufnimmt, bleibt der Fehler NO_SSM_INSTANCE_ROLE für die Anwendung bestehen.
Neue Komponenten müssen möglicherweise konfiguriert werden.	Application Insights erkennt, dass es neue Komponenten in der Anwendungsressourcengruppe gibt. Um dies zu beheben, konfigurieren Sie die neuen Komponenten entsprechend.	–

Von Amazon CloudWatch Application Insights unterstützte Protokolle und Metriken

Die folgenden Listen zeigen die unterstützten Protokolle und Metriken für Amazon CloudWatch Application Insights.

CloudWatch Application Insights unterstützt die folgenden Protokolle:

- Microsoft IIS-Protokolle (Internet Information Services)
- Fehlerprotokoll für SQL Server in EC2
- Benutzerdefinierte .NET-Anwendungsprotokolle, wie z. B. Log4Net
- Windows-Ereignisprotokolle, einschließlich Windows-Protokolle (System-, Anwendungs- und Sicherheitsprotokolle) und Anwendungen- sowie Services-Protokolle
- Amazon CloudWatch Logs für AWS Lambda
- Fehler- und Slow Query-Protokoll für RDS MySQL, Aurora MySQL und MySQL auf EC2
- Postgresql-Protokoll für PostgreSQL RDS und PostgreSQL auf EC2
- Amazon CloudWatch Logs für AWS Step Functions
- Ausführungsprotokolle und Zugriffsprotokolle (JSON, CSV und XML, aber nicht CLF) für REST-API-Phasen von API Gateway
- Prometheus-JMX-Exporter-Protokolle (EMF)
- Warnungsprotokolle und Listener-Protokolle für Oracle auf Amazon RDS und Oracle auf Amazon EC2
- Container-Logs werden von Amazon ECS-Containern an CloudWatch den `awslogs`Log-Treiber weitergeleitet.
- Weiterleitung von Container-Protokollen von Amazon ECS-Containern an den CloudWatch FireLens Container-Log-Router.
- Routing von Container-Protokollen von Amazon EKS oder Kubernetes, die auf Amazon EC2 ausgeführt werden, an den Fluent Bit- oder Fluentd-Protokollprozessor CloudWatch mit Container Insights.
- SAP HANA – Nachverfolgung und Fehlerprotokolle
- HA-Pacemaker-Protokolle
- SAP-ASE-Serverprotokolle
- SAP-ASE-Backup-Serverprotokolle

- SAP-ASE-Replikationsserverprotokolle
- SAP-ASE-RMA-Agentenprotokolle
- SAP-ASE-Fault-Manager-Protokolle
- NetWeaver Ablaufverfolgungsprotokolle für SAP-Entwickler
- Verarbeiten Sie Metriken für Windows-Prozesse mithilfe des Proctstat-Plug-ins für den Agenten CloudWatch
- Öffentliche DNS-Abfrageprotokolle für die gehostete Zone
- Amazon Route 53 Resolver DNS-Abfrageprotokolle

CloudWatch Application Insights unterstützt die folgenden Protokollklassen:

- Standard — Amazon CloudWatch Application Insights erfordert, dass Protokollgruppen mit der Protokollklasse CloudWatch Logs Standard konfiguriert werden, um die Überwachung zu ermöglichen.

CloudWatch Application Insights unterstützt Metriken für die folgenden Anwendungskomponenten:

- Amazon Elastic Compute Cloud (EC2)
  - CloudWatch integrierte Metriken
  - CloudWatch Agent-Metriken (Windows-Server)
  - CloudWatch Metriken für Agentenprozesse (Windows-Server)
  - CloudWatch Agent-Metriken (Linux-Server)
- Elastic Block Store (EBS)
- Amazon Elastic File System (Amazon EFS)
- Elastic Load Balancer (ELB)
- Application ELB
- Amazon EC2 Auto-Scaling-Gruppen
- Amazon Simple Queue Server (SQS)
- Amazon Relational Database Service (RDS)
  - RDS-Datenbank-Instances
  - RDS-Datenbank-Cluster
- AWS Lambda Funktion
- Amazon-DynamoDB-Tabelle.

- Amazon-S3-Bucket
- AWS Step Functions
  - Execution-level
  - Aktivität
  - Lambda-Funktion
  - Service-Integration
  - Step Functions API
- API-Gateway-REST-API-Phasen
- SAP HANA
- SAP ASE
- SAP ASE bei Amazon EC2 mit hoher Verfügbarkeit
- SAP NetWeaver
- HA-Cluster
- Java
- Amazon Elastic Container Service (Amazon ECS)
  - CloudWatch integrierte Metriken
  - Container-Insights-Metriken
  - Container-Insights-Prometheus-Metriken
- Kubernetes auf AWS
  - Container-Insights-Metriken
  - Container-Insights-Prometheus-Metriken
- Amazon FSx
- Amazon VPC
- Amazon-VPC-NAT-Gateways
- Amazon-Route-53-Zustandsprüfung
- Gehostete Zone von Amazon Route 53
- Amazon Route 53 Resolver Endpunkt
- AWS Network Firewall Regelgruppe
- AWS Network Firewall Zuordnung von Regelgruppen
- Metriken mit Datenpunktanforderungen

- AWS/ApplicationELB
- AWS/ AutoScaling
- AWS/EC2
- Elastic Block Store (EBS)
- AWS/ELB
- AWS/RDS
- AWS/Lambda
- AWS/SQS
- AWS/CWAgent
- AWS/DynamoDB
- AWS/S3
- AWS/Zustände
- AWS/ ApiGateway
- AWS/SNS

- Empfohlene Metriken
- Leistungsindikator-Metriken

Amazon Elastic Compute Cloud (EC2)

CloudWatch Application Insights unterstützt die folgenden Metriken:

Metriken

- CloudWatch integrierte Metriken
- CloudWatch Agent-Metriken (Windows-Server)
- CloudWatch Metriken für Agentenprozesse (Windows-Server)
- CloudWatch Agent-Metriken (Linux-Server)

CloudWatch integrierte Metriken

CPU CreditBalance

Zentralprozessor CreditUsage

Zentralprozessor SurplusCreditBalance

Zentralprozessor SurplusCreditsCharged

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

EBS% ByteBalance

EBSIOBalance%

EBS ReadBytes

EBS ReadOps

EBS WriteBytes

EBS WriteOps

NetworkIn

NetworkOut

NetworkPacketsIn

NetworkPacketsOut

StatusCheckFailed

StatusCheckFailed_Instanz

StatusCheckFailed_System

CloudWatch Agent-Metriken (Windows-Server)

.NET CLR-Ausnahmenanzahl der ausgelösten Ausnahmen

.NET CLR-Ausnahmenanzahl der ausgelösten Ausnahmen/Sek.

.NET CLR-Ausnahmenanzahl der Filter/Sek.

.NET-CLR-Ausnahmeanzahl der Finallys/Sek.

.NET CLR-Ausnahmen zur Erfassung der Tiefe/Sek.

.NET CLR-Interop-Anzahl der CCWs

.NET CLR Interop-Anzahl der Stubs

.NET CLR-Interop-Anzahl der TLB-Exporte/Sek.

.NET CLR-Interop-Anzahl der TLB-Importe/Sek.

.NET CLR Interop-Anzahl Marshaling

.NET CLR Jit Zeitprozentsatz in Jit

.NET CLR Jit Standard-Jit-Fehler

.NET CLR Laden Zeitprozentsatz für Laden

.NET CLR-Laderate von Ladefehlern

LocksAndThreads .NET-CLR-Konfliktrate/Sekunde

.NET CLR-Warteschlangenlänge/Sekunde LocksAndThreads

.NET CLR Speicher Anzahl der insges. best. Bytes

.NET CLR-Arbeitsspeicherzeit in % in GC

.NET CLR Networking 4.0.0.0 Durchschnittliche Warteschlangenzeit HttpWebRequest

.NET CLR Networking 4.0.0.0 abgebrochen/Sek HttpWebRequests

.NET CLR Networking 4.0.0.0 ist HttpWebRequests ausgefallen/Sek

.NET CLR Networking 4.0.0.0 in Warteschlangen/Sek HttpWebRequests

APP_POOL_WAS Ping-Fehler insgesamt beim Worker-Prozess

ASP.NET Anwendungsneustarts

ASP.NET-Anwendungen% verwaltete Prozessorzeit (geschätzt)

ASP.NET-Anwendungsfehler insgesamt/s

ASP.NET-Anwendungsfehler, die während der Ausführung nicht verarbeitet werden/s

ASP.NET-Anwendungsanforderungen in der Anwendungswarteschlange

ASP.NET-Anwendungen – Anforderungen/s

ASP.NET Anforderungswartezeit

ASP.NET Anforderungen in Warteschlange

Warteschlangen für HTTP-Dienstanforderungen CurrentQueueSize

LogicalDisk % freier Speicherplatz

Speicher % übertragene Bytes im Gebrauch

Verfügbarer Speicher Mbytes

Speicherseiten/Sek.

Netzwerkschnittstellen-Bytes gesamt/Sekunde

Auslagerungsdatei % Verwendung

PhysicalDisk % Festplattenzeit

PhysicalDisk Durchschn. Länge der Datenträgerwarteschlange

PhysicalDisk Durchschn. Festplatte Sek/Lesen

PhysicalDisk Durchschn. Disk sec/Write

PhysicalDisk Gelesene Festplatten-Bytes/Sek

PhysicalDisk Lesevorgänge auf der Festplatte pro Sekunde

PhysicalDisk Bytes/Sekunde beim Schreiben auf die Festplatte

PhysicalDisk Schreibvorgänge auf der Festplatte pro Sekunde

Processor % Idle Time

Processor % Interrupt Time

Processor % Processor Time

Processor % User Time

SQLServer:Access Methods Forwarded Records/sec

SQLServer:Access Methods Full Scans/sec

SQLServer:Access Methods Page Splits/sec

SQLServer:Buffer Manager Buffer cache hit ratio

SQLServer:Buffer Manager Page life expectancy

SQLServer:General Statistics Processes blocked

SQLServer:General Statistics User Connections

SQLServer:Latches Average Latch Wait Time (ms)

SQLServer:Locks Average Wait Time (ms)

SQLServer:Locks Lock Timeouts/sec

SQLServer:Locks Lock Waits/sec

SQLServer:Locks Number of Deadlocks/sec

SQLServer:Memory Manager Memory Grants Pending

SQLServer:SQL Statistics Batch Requests/sec

SQLServer:SQL Statistics SQL Compilations/sec

SQLServer:SQL Statistics SQL Re-Compilations/sec

System Processor Queue Length

TCPv4 Connections Established

TCPv6 Connections Established

W3SVC_W3WP Dateicache-Leerungen

W3SVC_W3WP Datei-Cache-Fehlschläge

W3SVC_W3WP Anforderungen/Sek.

W3SVC_W3WP URI-Cache-Leerungen

W3SVC_W3WP URI-Cache-Fehlschläge

Empfangene Web-Service-Bytes/s

Gesendete Web-Service-Bytes/s

Web-Service-Verbindungsversuche/Sek.

Aktuelle Webservice-Verbindungen

Web-Service-Abrufanforderungen/Sek.

Web-Service-Beitragsanforderungen/Sek.

Empfangene Bytes/Sek.

Länge der normalen Nachrichtenwarteschlange/Sek.

Länge der Warteschlange für dringende Nachrichten/Sek.

Anzahl der erneuten Verbindung

Länge der Warteschlange für unbestätigte Nachrichten/Sek.

Ausstehende Nachrichten

Gesendete Meldungen/Sek.

Datenbank-Aktualisierungsnachrichten/Sek.

Aktualisieren von Nachrichten/Sek.

Bereinigungen/Sek.

Gespeicherte Krypto-Prüfpunkte/Sek.

Wiederhergestellte Krypto-Prüfpunkte/Sek

Wiederhergestellte Registrierungs-Prüfpunkte/Sek.

Gespeicherte Registrierungs-Prüfpunkte/Sek.

Cluster-API-Aufrufe/Sek.

Ressourcen-API-Aufrufe/Sek.

Cluster-Handles/Sek.

Ressourcen-Handles/Sek.

CloudWatch Metriken für Agentenprozesse (Windows-Server)

Prozessmetriken werden mithilfe des Procstat-Plug-ins für CloudWatch Agenten erfasst. Nur Amazon-EC2-Instances, auf denen Windows-Workloads ausgeführt werden, unterstützen Prozessmetriken.

procstat cpu_time_system

procstat cpu_time_user

procstat cpu_usage

procstat memory_rss

procstat memory_vms

procstat read_bytes

procstat write_bytes

.procstat read_count

procstat write_count

CloudWatch Agent-Metriken (Linux-Server)

cpu_time_active

cpu_time_guest

cpu_time_guest_nice

cpu_time_idle

cpu_time_iowait

cpu_time_irq

cpu_time_nice

cpu_time_softirq

cpu_time_steal

cpu_time_system

cpu_time_user

cpu_usage_active

cpu_usage_guest

cpu_usage_guest_nice

cpu_usage_idle

cpu_usage_iowait

cpu_usage_irq

cpu_usage_nice

cpu_usage_softirq

cpu_usage_steal

cpu_usage_system

cpu_usage_user

disk_free

disk_inodes_free

disk_inodes_used

disk_used

disk_used_percent

diskio_io_time

diskio_iops_in_progress

diskio_read_bytes

diskio_read_time

diskio_reads

diskio_write_bytes

diskio_write_time

diskio_writes

mem_active

mem_available

mem_available_percent

mem_buffered

mem_cached

mem_free

mem_inactive

mem_used

mem_used_percent

net_bytes_recv

net_bytes_sent

net_drop_in

net_drop_out

net_err_in

net_err_out

net_packets_recv

net_packets_sent

netstat_tcp_close

netstat_tcp_close_wait

netstat_tcp_closing

netstat_tcp_established

netstat_tcp_fin_wait1

netstat_tcp_fin_wait2

netstat_tcp_last_ack

netstat_tcp_listen

netstat_tcp_none

netstat_tcp_syn_recv

netstat_tcp_syn_sent

netstat_tcp_time_wait

netstat_udp_socket

processes_blocked

processes_dead

processes_idle

processes_paging

processes_running

processes_sleeping

processes_stopped

processes_total

processes_total_threads

processes_wait

processes_zombies

swap_free

swap_used

swap_used_percent

Elastic Block Store (EBS)

CloudWatch Application Insights unterstützt die folgenden Metriken:

VolumeReadBytes

VolumeWriteBytes

VolumeReadOps

VolumeWriteOps

VolumeTotalReadTime

VolumeTotalWriteTime

VolumeIdleTime

VolumeQueueLength

VolumeThroughputPercentage

VolumeConsumedReadWriteOps

BurstBalance

Amazon Elastic File System (Amazon EFS)

CloudWatch Application Insights unterstützt die folgenden Metriken:

BurstCreditBalance

PercentIOLimit

PermittedThroughput

MeteredIOBytes

TotalIOBytes

DataWriteIOBytes

DataReadIO-Bytes

MetadataIOBytes

ClientConnections

TimeSinceLastSync

StorageBytes

Durchsatz

PercentageOfPermittedThroughputUtilization

ThroughputIOPS

PercentThroughputDataReadIOByte

PercentThroughputDataWriteIOBytes

PercentageOfIOPS IOBytes DataRead

PercentageOfIOPS DataWrite IOBytes

AverageDataReadIO BytesSize

AverageDataWriteIO BytesSize

Elastic Load Balancer (ELB)

CloudWatch Application Insights unterstützt die folgenden Metriken:

Geschätztes ALB ActiveConnectionCount

EstimatedALBConsumedLCUs

Geschätztes ALB NewConnectionCount

EstimatedProcessedBytes

HTTPCode_Backend_4XX

HTTPCode_Backend_5XX

HealthyHostCount

RequestCount

UnHealthyHostCount

Application ELB

CloudWatch Application Insights unterstützt die folgenden Metriken:

Geschätztes ALB ActiveConnectionCount

EstimatedALBConsumedLCUs

Geschätztes ALB NewConnectionCount

EstimatedProcessedBytes

HTTPCode_Backend_4XX

HTTPCode_Backend_5XX

HealthyHostCount

Latency

RequestCount

SurgeQueueLength

UnHealthyHostCount

Amazon EC2 Auto-Scaling-Gruppen

CloudWatch Application Insights unterstützt die folgenden Metriken:

CPU CreditBalance

Zentralprozessor CreditUsage

Zentralprozessor SurplusCreditBalance

Zentralprozessor SurplusCreditsCharged

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

EBS% ByteBalance

EBSIOBalance%

EBS ReadBytes

EBS ReadOps

EBS WriteBytes

EBS WriteOps

NetworkIn

NetworkOut

NetworkPacketsIn

NetworkPacketsOut

StatusCheckFailed

StatusCheckFailed_Instanz

StatusCheckFailed_System

Amazon Simple Queue Server (SQS)

CloudWatch Application Insights unterstützt die folgenden Metriken:

ApproximateAgeOfOldestMessage

ApproximateNumberOfMessagesDelayed

ApproximateNumberOfMessagesNotVisible

ApproximateNumberOfMessagesVisible

NumberOfEmptyReceives

NumberOfMessagesDeleted

NumberOfMessagesReceived

NumberOfMessagesSent

Amazon Relational Database Service (RDS)

CloudWatch Application Insights unterstützt die folgenden Metriken:

Metriken

- RDS-Datenbank-Instances
- RDS-Datenbank-Cluster

RDS-Datenbank-Instances

BurstBalance

CPU CreditBalance

CPUUtilization

DatabaseConnections

DiskQueueDepth

SQL ist fehlgeschlagen ServerAgentJobsCount

FreeStorageSpace

FreeableMemory

NetworkReceiveThroughput

NetworkTransmitThroughput

ReadIOPS

ReadLatency

ReadThroughput

WriteIOPS

WriteLatency

WriteThroughput

RDS-Datenbank-Cluster

ActiveTransactions

AuroraBinlogReplicaLag

AuroraReplicaLag

BackupRetentionPeriodStorageUsed

BinLogDiskUsage

BlockedTransactions

BufferCacheHitRatio

CPUUtilization

CommitLatency

CommitThroughput

DDLLatency

DDLThroughput

DMLLatency

DMLThroughput

DatabaseConnections

Deadlocks

DeleteLatency

DeleteThroughput

EngineUptime

FreeLocalStorage

FreeableMemory

InsertLatency

InsertThroughput

LoginFailures

NetworkReceiveThroughput

NetworkThroughput

NetworkTransmitThroughput

Abfragen

ResultSetCacheHitRatio

SelectLatency

SelectThroughput

SnapshotStorageUsed

TotalBackupStorageBilled

UpdateLatency

UpdateThroughput

VolumeBytesUsed

VolumeReadIOPs

VolumeWriteIOPs

AWS Lambda Funktion

CloudWatch Application Insights unterstützt die folgenden Metriken:

Fehler

DeadLetterErrors

Dauer

Drosselungen

IteratorAge

ProvisionedConcurrencySpilloverInvocations

Amazon-DynamoDB-Tabelle.

CloudWatch Application Insights unterstützt die folgenden Metriken:

SystemErrors

UserErrors

ConsumedReadCapacityUnits

ConsumedWriteCapacityUnits

ReadThrottleEvents

WriteThrottleEvents

TimeToLiveDeletedItemCount

ConditionalCheckFailedRequests

TransactionConflict

ReturnedRecordsCount

PendingReplicationCount

ReplicationLatency

Amazon-S3-Bucket

CloudWatch Application Insights unterstützt die folgenden Metriken:

ReplicationLatency

BytesPendingReplication

OperationsPendingReplication

4xxErrors

5xxErrors

AllRequests

GetRequests

PutRequests

DeleteRequests

HeadRequests

PostRequests

SelectRequests

ListRequests

SelectScannedBytes

SelectReturnedBytes

FirstByteLatency

TotalRequestLatency

BytesDownloaded

BytesUploaded

AWS Step Functions

CloudWatch Application Insights unterstützt die folgenden Metriken:

Metriken

- Execution-level
- Aktivität
- Lambda-Funktion
- Service-Integration
- Step Functions API

Execution-level

ExecutionTime

ExecutionThrottled

ExecutionsFailed

ExecutionsTimedOut

ExecutionsAborted

ExecutionsSucceeded

ExecutionsStarted

Aktivität

ActivityRunTime

ActivityScheduleTime

ActivityTime

ActivitiesFailed

ActivitiesHeartbeatTimedOut

ActivitiesTimedOut

ActivitiesScheduled

ActivitiesSucceeded

ActivitiesStarted

Lambda-Funktion

LambdaFunctionRunTime

LambdaFunctionScheduleTime

LambdaFunctionTime

LambdaFunctionsFailed

LambdaFunctionsTimedOut

LambdaFunctionsScheduled

LambdaFunctionsSucceeded

LambdaFunctionsStarted

Service-Integration

ServiceIntegrationRunTime

ServiceIntegrationScheduleTime

ServiceIntegrationTime

ServiceIntegrationsFailed

ServiceIntegrationsTimedOut

ServiceIntegrationsScheduled

ServiceIntegrationsSucceeded

ServiceIntegrationsStarted

Step Functions API

ThrottledEvents

ProvisionedBucketSize

ProvisionedRefillRate

ConsumedCapacity

API-Gateway-REST-API-Phasen

CloudWatch Application Insights unterstützt die folgenden Metriken:

4XXError

5XXError

IntegrationLatency

Latency

CacheHitCount

CacheMissCount

SAP HANA

> **Note**
>
> CloudWatch Application Insights unterstützt nur einzelne SID-HANA-Umgebungen. Wenn mehrere HANA-SIDs angeschlossen sind, wird die Überwachung nur für die erste erkannte SID eingerichtet.

CloudWatch Application Insights unterstützt die folgenden Metriken:

hanadb_every_service_started_status

hanadb_daemon_service_started_status

hanadb_preprocessor_service_started_status

hanadb_webdispatcher_service_started_status

hanadb_compileserver_service_started_status

hanadb_nameserver_service_started_status

hanadb_server_startup_time_variations_seconds

hanadb_level_5_alerts_count

hanadb_level_4_alerts_count

hanadb_out_of_memory_events_count

hanadb_max_trigger_read_ratio_percent

hanadb_max_trigger_write_ratio_percent

hanadb_log_switch_wait_ratio_percent

hanadb_log_switch_race_ratio_percent

hanadb_time_since_last_savepoint_seconds

hanadb_disk_usage_highlevel_percent

hanadb_max_converter_page_number_count

hanadb_long_running_savepoints_count

hanadb_failed_io_reads_count

hanadb_failed_io_writes_count

hanadb_disk_data_unused_percent

hanadb_current_allocation_limit_used_percent

hanadb_table_allocation_limit_used_percent

hanadb_host_total_physical_memory_mb

hanadb_host_physical_memory_used_mb

hanadb_host_physical_memory_free_mb

hanadb_swap_memory_free_mb

hanadb_swap_memory_used_mb

hanadb_host_allocation_limit_mb

hanadb_host_total_memory_used_mb

hanadb_host_total_peak_memory_used_mb

hanadb_host_total_allocation_limit_mb

hanadb_host_code_size_mb

hanadb_host_shared_memory_allocation_mb

hanadb_cpu_usage_percent

hanadb_cpu_user_percent

hanadb_cpu_system_percent

hanadb_cpu_waitio_percent

hanadb_cpu_busy_percent

hanadb_cpu_idle_percent

hanadb_long_delta_merge_count

hanadb_unsuccessful_delta_merge_count

hanadb_successful_delta_merge_count

hanadb_row_store_allocated_size_mb

hanadb_row_store_free_size_mb

hanadb_row_store_used_size_mb

hanadb_temporary_tables_count

hanadb_large_non_compressed_tables_count

hanadb_total_non_compressed_tables_count

hanadb_longest_running_job_seconds

hanadb_average_commit_time_milliseconds

hanadb_suspended_sql_statements_count

hanadb_plan_cache_hit_ratio_percent

hanadb_plan_cache_lookup_count

hanadb_plan_cache_hit_count

hanadb_plan_cache_total_execution_microseconds

hanadb_plan_cache_cursor_duration_microseconds

hanadb_plan_cache_preparation_microseconds

hanadb_plan_cache_evicted_count

hanadb_plan_cache_evicted_microseconds

hanadb_plan_cache_evicted_preparation_count

hanadb_plan_cache_evicted_execution_count

hanadb_plan_cache_evicted_preparation_microseconds

hanadb_plan_cache_evicted_cursor_duration_microseconds

hanadb_plan_cache_evicted_total_execution_microseconds

hanadb_plan_cache_evicted_plan_size_mb

hanadb_plan_cache_count

hanadb_plan_cache_preparation_count

hanadb_plan_cache_execution_count

hanadb_network_collision_rate

hanadb_network_receive_rate

hanadb_network_transmit_rate

hanadb_network_packet_receive_rate

hanadb_network_packet_transmit_rate

hanadb_network_transmit_error_rate

hanadb_network_receive_error_rate

hanadb_time_until_license_expires_days

hanadb_is_license_valid_status

hanadb_local_running_connections_count

hanadb_local_idle_connections_count

hanadb_remote_running_connections_count

hanadb_remote_idle_connections_count

hanadb_last_full_data_backup_age_days

hanadb_last_data_backup_age_days

hanadb_last_log_backup_age_hours

hanadb_failed_data_backup_past_7_days_count

hanadb_failed_log_backup_past_7_days_count

hanadb_oldest_backup_in_catalog_age_days

hanadb_backup_catalog_size_mb

hanadb_hsr_replication_status

hanadb_hsr_log_shipping_delay_seconds

hanadb_hsr_secondary_failover_count

hanadb_hsr_secondary_reconnect_count

hanadb_hsr_async_buffer_used_mb

hanadb_hsr_secondary_active_status

hanadb_handle_count

hanadb_ping_time_milliseconds

hanadb_connection_count

hanadb_internal_connection_count

hanadb_external_connection_count

hanadb_idle_connection_count

hanadb_transaction_count

hanadb_internal_transaction_count

hanadb_external_transaction_count

hanadb_user_transaction_count

hanadb_blocked_transaction_count

hanadb_statement_count

hanadb_active_commit_id_range_count

hanadb_mvcc_version_count

hanadb_pending_session_count

hanadb_record_lock_count

hanadb_read_count

hanadb_write_count

hanadb_merge_count

hanadb_unload_count

hanadb_active_thread_count

hanadb_waiting_thread_count

hanadb_total_thread_count

hanadb_active_sql_executor_count

hanadb_waiting_sql_executor_count

hanadb_total_sql_executor_count

hanadb_data_write_size_mb

hanadb_data_write_time_milliseconds

hanadb_log_write_size_mb

hanadb_log_write_time_milliseconds

hanadb_data_read_size_mb

hanadb_data_read_time_milliseconds

hanadb_log_read_size_mb

hanadb_log_read_time_milliseconds

hanadb_data_backup_write_size_mb

hanadb_data_backup_write_time_milliseconds

hanadb_log_backup_write_size_mb

hanadb_log_backup_write_time_milliseconds

hanadb_mutex_collision_count

hanadb_read_write_lock_collision_count

hanadb_admission_control_admit_count

hanadb_admission_control_reject_count

hanadb_admission_control_queue_size_mb

hanadb_admission_control_wait_time_milliseconds

SAP ASE

CloudWatch Application Insights unterstützt die folgenden Metriken:

asedb_database_availability

asedb_trunc_log_on_chkpt_enabled

asedb_last_db_backup_age_in_days

asedb_last_transaction_log_backup_age_in_hours

asedb_suspected_database

asedb_db_space_usage_percent

asedb_db_log_space_usage_percent

asedb_locked_login

asedb_has_mixed_log_and_data

asedb_runtime_for_open_transactions

asedb_data_cache_hit_ratio

asedb_data_cache_usage

asedb_sql_cache_hit_ratio

asedb_cache_usage

asedb_run_queue_length

asedb_number_of_rollbacks

asedb_number_of_commits

asedb_number_of_transactions

asedb_outstanding_disk_io

asedb_percent_io_busy

asedb_percent_system_busy

asedb_percent_locks_active

asedb_scheduled_jobs_failed_percent

asedb_user_connections_percent

asedb_query_logical_reads

asedb_query_physical_reads

asedb_query_cpu_time

asedb_query_memory_usage

SAP ASE bei Amazon EC2 mit hoher Verfügbarkeit

CloudWatch Application Insights unterstützt die folgenden Metriken:

asedb_ha_replication_state

asedb_ha_replication_mode

asedb_ha_replication_latency_in_minutes

SAP NetWeaver

CloudWatch Application Insights unterstützt die folgenden Kennzahlen:

Metrik	Beschreibung
sap_alerts_ ResponseTime	Der SAP-Antwortzeit-Alert von CCMS (RZ20) >R3Services>Dialog>. ResponseTime
sap_alerts_ ResponseTimeDialog	Der SAP-Antwortzeit-Dialog von CCMS (RZ20) >R3Services>Dialog>. ResponseTimeDialog
ResponseTimeDialogsap_alerts_ RFC	Der SAP-Antwortzeit-Alert von CCMS (RZ20) >R3Services> Dialog> RFC. ResponseT imeDialog
SAP_Alerts_DB RequestTime	Der SAP-Antwortzeit-Alert von CCMS (RZ20) >R3Services>Dialog>DB. RequestTime
sap_alerts_ FrontendResponseTime	Der SAP-Antwortzeit-Alert von CCMS (RZ20) >R3Services > Dialog>. FrontEndResponseTi me
sap_alerts_Database	Das SAP-System hat datenbankbezogene Fehler protokolliert. Warnung von SM21 oder CCMS (RZ20) >R3Syslog>Database.
sap_alerts_ QueueTime	Der SAP-Warteschlangenzeitalarm von CCMS (RZ20) >R3Services>Dialog>. QueueTime
sap_alerts_ AbortedJobs	Fehlgeschlagene Hintergrundaufträge im SAP-System. Warnung von (RZ20) > R3-Dienste > Hintergrund >. AbortedJobs
sap_alerts_ BasisSystem	Das SAP-System hat Fehler auf Systemebene protokolliert. Warnung von SM21 oder CCMS (RZ20) >R3Syslog>. BasisSystem

Metrik	Beschreibung
sap_alerts_Security	Das SAP-System protokollierte sicherhei tsrelevante Meldungen. Warnung von SM21 oder CCMS (RZ20)>R3Syslog>Security.
sap_alerts_System	Das SAP-System hat sicherheits- oder auditbezogene Nachrichten protokolliert. Warnung von SM21 oder CCMS (RZ20)>Se curity>System.
sap_alerts_ LongRunners	In Ihrem SAP-System gibt es Programme mit langer Laufzeit. Warnung von CCMS (RZ20) >R3Services > Dialog>. LongRunners
sap_alerts_ SqlError	Es gibt Fehlerprotokolle auf der SAP-Daten bank-Client-Ebene. Warnung vom CCMS (RZ20) > > >. DatabaseClient AbapSql SqlError
sap_alerts_State	Zustandsalarm von CCMS (RZ20)>OS Collector>State.
sap_alerts_Shortdumps	Shortdumps-Alarm von ST22 und CCMS (RZ20)>R3Abap>Shortdumps.
sap_alerts_Availability	Verfügbarkeitswarnung für die SAP-Anwen dungsserverinstanz von SM21, SM50, SM51, SM66 und CCMS (RZ20) > >Verfügbarkeit. InstanceAsTask
sap_dispatcher_queue_high	Die Funktion `GetQueueStatistic` des SAPControl-Web-Services liefert die höchste Anzahl der Dispatcher-Warteschlangen.
sap_dispatcher_queue_max	Die Funktion `GetQueueStatistic` des SAPControl-Web-Services liefert die maximale Anzahl der Dispatcher-Warteschlangen.

Metrik	Beschreibung
sap_dispatcher_queue_now	Die Funktion `GetQueueStatistic` des SAPControl-Web-Services liefert die jetzige Anzahl der Dispatcher-Warteschlangen.
sap_dispatcher_queue_reads	Die Funktion `GetQueueStatistic` des SAPControl-Web-Services liefert die Anzahl der Dispatcher-Warteschlangen.
sap_dispatcher_queue_writes	Die SAPControl-Webservice-Funktion `GetQueueStatistic` gibt die Anzahl der Schreibvorgänge in der Dispatcher-Wartesc hlange an.
sap_enqueue_server_arguments_high	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert die höchste Anzahl der Enqueue-Argumente.
sap_enqueue_server_arguments_max	Die SAPControl-Webservice-Funktion `EnqGetStatistic` stellt die Enqueue-A rgumente auf maximal.
sap_enqueue_server_arguments_now	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert die maximale Anzahl der Enqueue-Argumente.
sap_enqueue_server_arguments_state	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert den Status der Enqueue-Argumente.
sap_enqueue_server_backup_requests	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert die Enqueue-B ackup-Anforderungen.
sap_enqueue_server_cleanup_requests	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert die Enqueue-C leanup-Anforderungen.

Metrik	Beschreibung
sap_enqueue_server_dequeue_all_requests	Die SAPControl-Webservice-Funktion `EnqGetStatistic` sorgt dafür, dass alle Anfragen aus der Warteschlange entfernt werden.
sap_enqueue_server_dequeue_errors	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert die Dequeue-Fehler.
sap_enqueue_server_dequeue_requests	Die SAPControl-Webservice-Funktion `EnqGetStatistic` stellt die Dequeue-Anforderungen bereit.
sap_enqueue_server_enqueue_errors	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert die Enqueue-Fehler.
sap_enqueue_server_enqueue_rejects	Die SAPControl-Webservice-Funktion `EnqGetStatistic` stellt die Enqueue-Ablehnungen zur Verfügung.
sap_enqueue_server_enqueue_requests	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert die Enqueue-Anforderungen.
sap_enqueue_server_lock_time	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert die Enqueue-Sperrzeit.
sap_enqueue_server_lock_wait_time	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert die Enqueue-Sperrwartezeit.
sap_enqueue_server_locks_high	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert die höchste Anzahl der Enqueue-Sperren.

Metrik	Beschreibung
sap_enqueue_server_locks_max	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert die maximale Anzahl der Enqueue-Sperren.
sap_enqueue_server_locks_now	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert die jetzige Anzahl der Enqueue-Sperren.
sap_enqueue_server_locks_state	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert den Status der Enqueue-Sperren.
sap_enqueue_server_owner_high	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert die Höchstzahl der Enqueue-Besitzer.
sap_enqueue_server_owner_max	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert das Maximum der Enqueue-Besitzer.
sap_enqueue_server_owner_now	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert die jetzigen Enqueue-Besitzer.
sap_enqueue_server_owner_state	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert den Status der Enqueue-Besitzer.
sap_enqueue_server_replication_state	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert den Status der Enqueue-Replikation.
sap_enqueue_server_reporting_requests	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert den Status der Berichtsanforderungen.

Metrik	Beschreibung
sap_enqueue_server_server_time	Die SAPControl-Webservice-Funktion `EnqGetStatistic` liefert die Enqueue-Serverzeit.
sap_HA_check_failover_config_state	Die Funktion SAPControl Web Service `HACheckFailoverConfig` stellt den SAP-Hochverfügbarkeitsstatus bereit.
sap_HA_get_failover_config_HAActive	Die SAPControl-Web-Service-Funktion `HAGetFailoverConfig` stellt die Konfiguration und den Status des Hochverfügbarkeits-Clusters von SAP bereit.
sap_start_service_processes	Die Funktion `GetProcessList` des SAPControl-Web-Service stellt den Status der Prozesse disp+work, IGS, gwrd, icman, Message Server und Enqueue-Server bereit.

HA-Cluster

CloudWatch Application Insights unterstützt die folgenden Metriken:

ha_cluster_pacemaker_stonith_enabled

ha_cluster_corosync_quorate

hanadb_webdispatcher_service_started_status

ha_cluster_pacemaker_nodes

ha_cluster_corosync_ring_errors

ha_cluster_pacemaker_fail_count

Java

CloudWatch Application Insights unterstützt die folgenden Metriken:

java_lang_memory_heapmemoryusage_used

java_lang_memory_heapmemoryusage_committed

java_lang_operatingsystem_openfiledescriptorcount

java_lang_operatingsystem_maxfiledescriptorcount

java_lang_operatingsystem_freephysicalmemorysize

java_lang_operatingsystem_freeswapspacesize

java_lang_threading_threadcount

java_lang_threading_daemonthreadcount

java_lang_classloading_loadedclasscount

java_lang_garbagecollector_collectiontime_copy

java_lang_garbagecollector_collectiontime_ps_scavenge

java_lang_garbagecollector_collectiontime_parnew

java_lang_garbagecollector_collectiontime_marksweepcompact

java_lang_garbagecollector_collectiontime_ps_marksweep

java_lang_garbagecollector_collectiontime_concurrentmarksweep

java_lang_garbagecollector_collectiontime_g1_young_generation

java_lang_garbagecollector_collectiontime_g1_old_generation

java_lang_garbagecollector_collectiontime_g1_mixed_generation

java_lang_operatingsystem_committedVirtualmemorysize

Amazon Elastic Container Service (Amazon ECS)

CloudWatch Application Insights unterstützt die folgenden Metriken:

Metriken

- CloudWatch integrierte Metriken
- Container-Insights-Metriken

- Container-Insights-Prometheus-Metriken

CloudWatch integrierte Metriken

CPUReservation

CPUUtilization

MemoryReservation

MemoryUtilization

GPUReservation

Container-Insights-Metriken

ContainerInstanceCount

CpuUtilized

CpuReserved

DeploymentCount

DesiredTaskCount

MemoryUtilized

MemoryReserved

NetworkRxBytes

NetworkTxBytes

PendingTaskCount

RunningTaskCount

ServiceCount

StorageReadBytes

StorageWriteBytes

TaskCount

TaskSetCount

instance_cpu_limit

instance_cpu_reserved_capacity

instance_cpu_usage_total

instance_cpu_utilization

instance_filesystem_utilization

instance_memory_limit

instance_memory_reserved_capacity

instance_memory_utilization

instance_memory_working_set

instance_network_total_bytes

instance_number_of_running_tasks

Container-Insights-Prometheus-Metriken

Java-JMX-Metriken

java_lang_memory_heapmemoryusage_used

java_lang_memory_heapmemoryusage_committed

java_lang_operatingsystem_openfiledescriptorcount

java_lang_operatingsystem_maxfiledescriptorcount

java_lang_operatingsystem_freephysicalmemorysize

java_lang_operatingsystem_freeswapspacesize

java_lang_threading_threadcount

java_lang_classloading_loadedclasscount

java_lang_threading_daemonthreadcount

java_lang_garbagecollector_collectiontime_copy

java_lang_garbagecollector_collectiontime_ps_scavenge

java_lang_garbagecollector_collectiontime_parnew

java_lang_garbagecollector_collectiontime_marksweepcompact

java_lang_garbagecollector_collectiontime_ps_marksweep

java_lang_garbagecollector_collectiontime_concurrentmarksweep

java_lang_garbagecollector_collectiontime_g1_young_generation

java_lang_garbagecollector_collectiontime_g1_old_generation

java_lang_garbagecollector_collectiontime_g1_mixed_generation

java_lang_operatingsystem_committedVirtualmemorysize

Kubernetes auf AWS

CloudWatch Application Insights unterstützt die folgenden Metriken:

Metriken

- Container-Insights-Metriken
- Container-Insights-Prometheus-Metriken

Container-Insights-Metriken

cluster_failed_node_count

cluster_node_count

namespace_number_of_running_pods

node_cpu_limit

node_cpu_reserved_capacity

node_cpu_usage_total

node_cpu_utilization

node_filesystem_utilization

node_memory_limit

node_memory_reserved_capacity

node_memory_utilization

node_memory_working_set

node_network_total_bytes

node_number_of_running_container

node_number_of_running_pods

pod_cpu_reserved_capacity

pod_cpu_utilization

pod_cpu_utilization_over_pod_limit

pod_memory_reserved_capacity

pod_memory_utilization

pod_memory_utilization_over_pod_limit

pod_network_rx_bytes

pod_network_tx_bytes

service_number_of_running_pods

Container-Insights-Prometheus-Metriken

Java-JMX-Metriken

java_lang_memory_heapmemoryusage_used

java_lang_memory_heapmemoryusage_committed

java_lang_operatingsystem_openfiledescriptorcount

java_lang_operatingsystem_maxfiledescriptorcount

java_lang_operatingsystem_freephysicalmemorysize

java_lang_operatingsystem_freeswapspacesize

java_lang_threading_threadcount

java_lang_classloading_loadedclasscount

java_lang_threading_daemonthreadcount

java_lang_garbagecollector_collectiontime_copy

java_lang_garbagecollector_collectiontime_ps_scavenge

java_lang_garbagecollector_collectiontime_parnew

java_lang_garbagecollector_collectiontime_marksweepcompact

java_lang_garbagecollector_collectiontime_ps_marksweep

java_lang_garbagecollector_collectiontime_concurrentmarksweep

java_lang_garbagecollector_collectiontime_g1_young_generation

java_lang_garbagecollector_collectiontime_g1_old_generation

java_lang_garbagecollector_collectiontime_g1_mixed_generation

java_lang_operatingsystem_committedVirtualmemorysize

Amazon FSx

CloudWatch Application Insights unterstützt die folgenden Metriken:

DataReadBytes

DataWriteBytes

DataReadOperations

DataWriteOperations

MetadataOperations

FreeStorageCapacity

FreeDataStorageCapacity

LogicalDiskUsage

PhysicalDiskUsage

Amazon VPC

CloudWatch Application Insights unterstützt die folgenden Metriken:

NetworkAddressUsage

NetworkAddressUsagePeered

VPC FirewallQueryVolume

Amazon-VPC-NAT-Gateways

CloudWatch Application Insights unterstützt die folgenden Metriken:

ErrorPortAllocation

IdleTimeoutCount

Amazon-Route-53-Zustandsprüfung

CloudWatch Application Insights unterstützt die folgenden Metriken:

ChildHealthCheckHealthyCount

ConnectionTime

HealthCheckPercentageHealthy

HealthCheckStatus

SSL HandshakeTime

TimeToFirstByte

Gehostete Zone von Amazon Route 53

CloudWatch Application Insights unterstützt die folgenden Metriken:

DNSQueries

DNSSEC InternalFailure

DNSSEC KeySigningKeysNeedingAction

DNSSEC KeySigningKeyMaxNeedingActionAge

DNSSEC KeySigningKeyAge

Amazon Route 53 Resolver Endpunkt

CloudWatch Application Insights unterstützt die folgenden Metriken:

EndpointHealthyENICount

EndpointUnHealthyEni zählen

InboundQueryVolume

OutboundQueryVolume

OutboundQueryAggregateVolume

AWS Network Firewall Regelgruppe

CloudWatch Application Insights unterstützt die folgenden Metriken:

FirewallRuleGroupQueryVolume

AWS Network Firewall Zuordnung von Regelgruppen

CloudWatch Application Insights unterstützt die folgenden Metriken:

FirewallRuleGroupVpcQueryVolume

Metriken mit Datenpunktanforderungen

Für Metriken ohne offensichtlichen Standardschwellenwert, bei denen ein Alarm aktiviert werden muss, wartet Application Insights, bis die Metrik genügend Datenpunkte hat, um einen

angemessenen Schwellenwert für den Alarm zu prognostizieren. Die Anforderungen an metrische Datenpunkte, die CloudWatch Application Insights überprüft, bevor ein Alarm ausgelöst wird, lauten wie folgt:

- Die Metrik hat mindestens 100 Datenpunkte von den letzten 15 bis 2 Tagen.
- Die Metrik hat mindestens 100 Datenpunkte vom letzten Tag.

Die folgenden Metriken erfüllen diese Datenpunktanforderungen. Beachten Sie, dass CloudWatch Agent-Metriken bis zu einer Stunde benötigen, um Alarme zu erstellen.

Metriken

- AWS/ApplicationELB
- AWS/ AutoScaling
- AWS/EC2
- Elastic Block Store (EBS)
- AWS/ELB
- AWS/RDS
- AWS/Lambda
- AWS/SQS
- AWS/CWAgent
- AWS/DynamoDB
- AWS/S3
- AWS/Zustände
- AWS/ ApiGateway
- AWS/SNS

AWS/ApplicationELB

ActiveConnectionCount

ConsumedLCUs

HTTPCode_ELB_4XX_Count

HTTPCode_Target_2XX_Count

HTTPCode_Target_3XX_Count

HTTPCode_Target_4XX_Count

HTTPCode_Target_5XX_Count

NewConnectionCount

ProcessedBytes

TargetResponseTime

UnHealthyHostCount

AWS/ AutoScaling

GroupDesiredCapacity

GroupInServiceInstances

GroupMaxSize

GroupMinSize

GroupPendingInstances

GroupStandbyInstances

GroupTerminatingInstances

GroupTotalInstances

AWS/EC2

ZENTRALPROZESSOR CreditBalance

Zentralprozessor CreditUsage

Zentralprozessor SurplusCreditBalance

Zentralprozessor SurplusCreditsCharged

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

EBS% ByteBalance

EBSIOBalance%

EBS ReadBytes

EBS ReadOps

EBS WriteBytes

EBS WriteOps

NetworkIn

NetworkOut

NetworkPacketsIn

NetworkPacketsOut

Elastic Block Store (EBS)

VolumeReadBytes

VolumeWriteBytes

VolumeReadOps

VolumeWriteOps

VolumeTotalReadTime

VolumeTotalWriteTime

VolumeIdleTime

VolumeQueueLength

VolumeThroughputPercentage

VolumeConsumedReadWriteOps

BurstBalance

AWS/ELB

Geschätztes ALB ActiveConnectionCount

EstimatedALBConsumedLCUs

Geschätztes ALB NewConnectionCount

EstimatedProcessedBytes

HTTPCode_Backend_4XX

HTTPCode_Backend_5XX

HealthyHostCount

Latency

RequestCount

SurgeQueueLength

UnHealthyHostCount

AWS/RDS

ActiveTransactions

AuroraBinlogReplicaLag

AuroraReplicaLag

BackupRetentionPeriodStorageUsed

BinLogDiskUsage

BlockedTransactions

Zentralprozessor CreditBalance

CommitLatency

CommitThroughput

DDLLatency

DDLThroughput

DMLLatency

DMLThroughput

DatabaseConnections

Deadlocks

DeleteLatency

DeleteThroughput

DiskQueueDepth

EngineUptime

FreeLocalStorage

FreeStorageSpace

FreeableMemory

InsertLatency

InsertThroughput

LoginFailures

NetworkReceiveThroughput

NetworkThroughput

NetworkTransmitThroughput

Abfragen

ReadIOPS

ReadThroughput

SelectLatency

SelectThroughput

SnapshotStorageUsed

TotalBackupStorageBilled

UpdateLatency

UpdateThroughput

VolumeBytesUsed

VolumeReadIOPs

VolumeWriteIOPs

WriteIOPS

WriteThroughput

AWS/Lambda

Fehler

DeadLetterErrors

Dauer

Drosselungen

IteratorAge

ProvisionedConcurrencySpilloverInvocations

AWS/SQS

ApproximateAgeOfOldestMessage

ApproximateNumberOfMessagesDelayed

ApproximateNumberOfMessagesNotVisible

ApproximateNumberOfMessagesVisible

NumberOfEmptyReceives

NumberOfMessagesDeleted

NumberOfMessagesReceived

NumberOfMessagesSent

AWS/CWAgent

LogicalDisk % Freier Speicherplatz

Speicher % übertragene Bytes im Gebrauch

Verfügbarer Speicher Mbytes

Netzwerkschnittstellen-Bytes gesamt/Sekunde

Auslagerungsdatei % Verwendung

PhysicalDisk % Festplattenzeit

PhysicalDisk Durchschn. Festplatte Sek/Lesen

PhysicalDisk Durchschn. Disk sec/Write

PhysicalDisk Gelesene Festplatten-Bytes/Sek

PhysicalDisk Lesevorgänge auf der Festplatte pro Sekunde

PhysicalDisk Bytes/Sekunde beim Schreiben auf die Festplatte

PhysicalDisk Schreibvorgänge auf der Festplatte pro Sekunde

Processor % Idle Time

Processor % Interrupt Time

Processor % Processor Time

Processor % User Time

SQLServer:Access Methods Forwarded Records/sec

SQLServer:Access Methods Page Splits/sec

SQLServer:Buffer Manager Buffer cache hit ratio

SQLServer:Buffer Manager Page life expectancy

SQLServer:Database Replica File Bytes Received/sec

SQLServer:Database Replica Log Bytes Received/sec

SQLServer:Database Replica Log remaining for undo

SQLServer:Database Replica Log Send Queue

SQLServer:Database Replica Mirrored Write Transaction/sec

SQLServer:Database Replica Recovery Queue

SQLServer:Database Replica Redo Bytes Remaining

SQLServer:Database Replica Redone Bytes/sec

SQLServer:Database Replica Total Log requiring undo

SQLServer:Database Replica Transaction Delay

SQLServer:General Statistics Processes blocked

SQLServer:SQL Statistics Batch Requests/sec

SQLServer:SQL Statistics SQL Compilations/sec

SQLServer:SQL Statistics SQL Re-Compilations/sec

System Processor Queue Length

TCPv4 Connections Established

TCPv6 Connections Established

AWS/DynamoDB

ConsumedReadCapacityUnits

ConsumedWriteCapacityUnits

ReadThrottleEvents

WriteThrottleEvents

TimeToLiveDeletedItemCount

ConditionalCheckFailedRequests

TransactionConflict

ReturnedRecordsCount

PendingReplicationCount

ReplicationLatency

AWS/S3

ReplicationLatency

BytesPendingReplication

OperationsPendingReplication

4xxErrors

5xxErrors

AllRequests

GetRequests

PutRequests

DeleteRequests

HeadRequests

PostRequests

SelectRequests

ListRequests

SelectScannedBytes

SelectReturnedBytes

FirstByteLatency

TotalRequestLatency

BytesDownloaded

BytesUploaded

AWS/Zustände

ActivitiesScheduled

ActivitiesStarted

ActivitiesSucceeded

ActivityScheduleTime

ActivityRuntime

ActivityTime

LambdaFunctionsScheduled

LambdaFunctionsStarted

LambdaFunctionsSucceeded

LambdaFunctionScheduleTime

LambdaFunctionRuntime

LambdaFunctionTime

ServiceIntegrationsScheduled

ServiceIntegrationsStarted

ServiceIntegrationsSucceeded

ServiceIntegrationScheduleTime

ServiceIntegrationRuntime

ServiceIntegrationTime

ProvisionedRefillRate

ProvisionedBucketSize

ConsumedCapacity

ThrottledEvents

AWS/ ApiGateway

4XXError

IntegrationLatency

Latency

DataProcessed

CacheHitCount

CacheMissCount

AWS/SNS

NumberOfNotificationsDelivered

NumberOfMessagesPublished

NumberOfNotificationsFailed

NumberOfNotificationsFilteredOut

NumberOfNotificationsFilteredOut-InvalidAttributes

NumberOfNotificationsFilteredOut-NoMessageAttributes

NumberOfNotificationsRedrivenToDlq

NumberOfNotificationsFailedToRedriveToDlq

SMS SuccessRate

Empfohlene Metriken

Die folgende Tabelle listet die empfohlenen Metriken für jeden Komponententyp auf.

Komponententyp	Workload-Typ	Empfohlene Metriken
EC2-Instance (Windows-Server)	Standard/Benutzerdefiniert	CPUUtilization StatusCheckFailed Processor % Processor Time Speicher % übertragene Bytes im Gebrauch LogicalDisk % Freier Speicherplatz Verfügbarer Speicher Mbytes
	Active Directory	CPUUtilization StatusCheckFailed Processor % Processor Time Speicher % übertragene Bytes im Gebrauch Verfügbarer Speicher Mbytes Datenbank ==> Instanzen Datenbank Cache % Treffer DirectoryServices Ausstehende DRA-Replikationsvorgänge DirectoryServices Ausstehende DRA-Replikationssynchronisationen

Komponententyp	Workload-Typ	Empfohlene Metriken
		Ausfall der rekursiven DNS-Abfrage/Sek LogicalDisk Durchschn. Länge der Datenträgerwarteschlange
	Java-Anwendung	CPUUtilization StatusCheckFailed Processor % Processor Time Speicher % übertragene Bytes im Gebrauch Verfügbarer Speicher Mbytes java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

Komponententyp	Workload-Typ	Empfohlene Metriken
	Microsoft IIS/.NET Web Front-End	CPUUtilization StatusCheckFailed Processor % Processor Time Speicher % übertragene Bytes im Gebrauch Verfügbarer Speicher Mbytes .NET CLR-Ausnahmenanzahl der ausgelösten Ausnahmen/Sek. .NET CLR Speicher Anzahl der insges. best. Bytes .NET CLR-Arbeitsspeicherzeit in % in GC ASP.NET-Anwendungsanforderungen in der Anwendungswarteschlange ASP.NET Anforderungen in Warteschlange ASP.NET Anwendung sneustarts

Komponententyp	Workload-Typ	Empfohlene Metriken
	Microsoft SQL Server Database Tier	CPUUtilization StatusCheckFailed Processor % Processor Time Speicher % übertragene Bytes im Gebrauch Verfügbarer Speicher Mbytes Auslagerungsdatei % Verwendung System Processor Queue Length Netzwerkschnittstellen-Bytes gesamt/Sekunde PhysicalDisk % Festplattenzeit SQLServer:Buffer Manager Buffer cache hit ratio SQLServer:Buffer Manager Page life expectancy SQLServer:General Statistics Processes blocked SQLServer:General Statistics User Connections SQLServer:Locks Number of Deadlocks/Sec SQLServer:SQL Statistics Batch Requests/sec

Komponententyp	Workload-Typ	Empfohlene Metriken
	MySQL	CPUUtilization StatusCheckFailed Processor % Processor Time Speicher % übertragene Bytes im Gebrauch LogicalDisk % Freier Speicherplatz Verfügbarer Speicher Mbytes
	.NET workerpool/Mid-Tier	CPUUtilization StatusCheckFailed Processor % Processor Time Speicher % übertragene Bytes im Gebrauch Verfügbarer Speicher Mbytes .NET CLR-Ausnahmenanzahl der ausgelösten Ausnahmen/Sek. .NET CLR Speicher Anzahl der insges. best. Bytes .NET CLR-Arbeitsspeicherzeit in % in GC

Komponententyp	Workload-Typ	Empfohlene Metriken
	.NET Core Tier	CPUUtilization StatusCheckFailed Processor % Processor Time Speicher % übertragene Bytes im Gebrauch Verfügbarer Speicher Mbytes
	Oracle	CPUUtilization StatusCheckFailed Processor % Processor Time Speicher % übertragene Bytes im Gebrauch LogicalDisk % Freier Speicherplatz Verfügbarer Speicher Mbytes
	Postgres	CPUUtilization StatusCheckFailed Processor % Processor Time Speicher % übertragene Bytes im Gebrauch LogicalDisk % Freier Speicherplatz Verfügbarer Speicher Mbytes

Komponententyp	Workload-Typ	Empfohlene Metriken
	SharePoint	CPUUtilization StatusCheckFailed Processor % Processor Time Speicher % übertragene Bytes im Gebrauch Verfügbarer Speicher Mbytes ASP.NET Anwendungs-Cache-API-Trim ASP.NET-Anforderungen abgelehnt ASP.NET Worker-Prozess wird neu gestartet Speicherseiten/Sek. SharePoint Cache veröffent lichen Cache leert den Cache// Sekunde SharePoint Ausführungszeit/ Seitenanforderung der Foundation SharePoint Festplattenbasiert er Cache Gesamtzahl der Cache-Komprimierungen SharePoint Trefferquote zwischen festplattenbasiertem Cache und Blob-Cache

Komponententyp	Workload-Typ	Empfohlene Metriken
		SharePoint Festplattenbasiert er Cache, Blob, Cache-Fül lrate SharePoint Festplattenbasiert er Cache: Blob, Cache-Lee rung//Sekunde ASP.NET Anforderungen in Warteschlange ASP.NET-Anwendungs anforderungen in der Anwendungswarteschlange ASP.NET Anwendung sneustarts LogicalDisk Durchschn. Disk sec/Write LogicalDisk Durchschn. Festplatte Sek/Lesen Processor % Interrupt Time
EC2-Instance (Linux-Server)	Standard/Benutzerdefiniert	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent

Komponententyp	Workload-Typ	Empfohlene Metriken
	Java-Anwendung	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize
	.NET Core-Ebene oder SQL Server-Datenbankebene	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent

Komponententyp	Workload-Typ	Empfohlene Metriken
	Oracle	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent
	Postgres	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent

Komponententyp	Workload-Typ	Empfohlene Metriken
EC2-Instance-Gruppe	SAP HANA (mehrere Knoten oder Einzelknoten)	• hanadb_server_startup_time_variations_seconds • hanadb_level_5_alerts_count • hanadb_level_4_alerts_count • hanadb_out_of_memory_events_count • hanadb_max_trigger_read_ratio_percent • hanadb_max_trigger_write_ratio_percent • hanadb_log_switch_race_ratio_percent • hanadb_time_since_last_savepoint_seconds • hanadb_disk_usage_highlevel_percent • hanadb_current_allocation_limit_used_percent • hanadb_table_allocation_limit_used_percent • hanadb_cpu_usage_percent • hanadb_plan_cache_hit_ratio_percent • hanadb_last_data_backup_age_days

Komponententyp	Workload-Typ	Empfohlene Metriken
EBS-Volume	Any	VolumeReadBytes VolumeWriteBytes VolumeReadOps VolumeWriteOps VolumeQueueLength VolumeThroughputPercentage VolumeConsumedReadWriteOps BurstBalance
Classic ELB	Any	HTTPCode_Backend_4XX HTTPCode_Backend_5XX Latency SurgeQueueLength UnHealthyHostCount
Application ELB	Any	HTTPCode_Target_4XX_Count HTTPCode_Target_5XX_Count TargetResponseTime UnHealthyHostCount

Komponententyp	Workload-Typ	Empfohlene Metriken
RDS-Datenbank-Instance	Any	CPUUtilization ReadLatency WriteLatency BurstBalance SQL ist fehlgeschlagen ServerAgentJobsCount
RDS-Datenbank-Cluster	Any	CPUUtilization CommitLatency DatabaseConnections Deadlocks FreeableMemory NetworkThroughput VolumeBytesUsed
Lambda-Funktion	Any	Dauer Fehler IteratorAge ProvisionedConcurrencySpillOverInvocations Drosselungen

Komponententyp	Workload-Typ	Empfohlene Metriken
SQS Queue	Any	ApproximateAgeOfOldestMessage ApproximateNumberOfMessagesVisible NumberOfMessagesSent
Amazon-DynamoDB-Tabelle.	Any	SystemErrors UserErrors ConsumedReadCapacityUnits ConsumedWriteCapacityUnits ReadThrottleEvents WriteThrottleEvents ConditionalCheckFailedRequests TransactionConflict

Komponententyp	Workload-Typ	Empfohlene Metriken
Amazon-S3-Bucket	Any	Wenn die Replikationskonfiguration mit Replication Time Control (RTC) aktiviert ist: ReplicationLatency BytesPendingReplication OperationsPendingReplication Wenn Anforderungsmetriken aktiviert sind: 5xxErrors 4xxErrors BytesDownloaded BytesUploaded

Komponententyp	Workload-Typ	Empfohlene Metriken
AWS Step Functions	Any	Allgemeines • ExecutionThrottled • ExecutionsAborted • ProvisionedBucketSize • ProvisionedRefillRate • ConsumedCapacity Wenn der Typ der Zustandsmaschine **EXPRESS** ist oder die Protokollgruppenebene **OFF** ist • ExecutionsFailed • ExecutionsTimedOut Wenn die Zustandsmaschine Lambda-Funktionen hat • LambdaFunctionsFailed • LambdaFunctionsTimedOut Wenn die Zustandsmaschine Aktivitäten hat • ActivitiesFailed • ActivitiesTimedOut • ActivitiesHeartbeatTimedOut Wenn die Zustandsmaschine Service-Integrationen hat • ServiceIntegrationsFailed

Komponententyp	Workload-Typ	Empfohlene Metriken
		• ServiceIntegrations TimedOut
API-Gateway-REST-API-Phase	Any	• 4xxErrors • 5xxErrors • Latency

Komponententyp	Workload-Typ	Empfohlene Metriken
ECS-Cluster	Any	CpuUtilized MemoryUtilized NetworkRxBytes NetworkTxBytes RunningTaskCount PendingTaskCount StorageReadBytes StorageWriteBytes CPUReservation (nur mit dem Starttyp EC2) CPUUtilization (nur mit dem Starttyp EC2) MemoryReservation (Nur EC2-Starttyp) MemoryUtilization (Nur EC2-Starttyp) GPU-Reservation (nur mit dem Starttyp EC2) instance_cpu_utilization (nur EC2-Starttyp) instance_filesystem_utilization (nur EC2-Starttyp) instance_memory_utilization (nur EC2-Starttyp)

Komponententyp	Workload-Typ	Empfohlene Metriken
		instance_network_total_bytes (nur EC2-Starttyp)

Komponententyp	Workload-Typ	Empfohlene Metriken
	Java-Anwendung	CpuUtilized MemoryUtilized NetworkRxBytes NetworkTxBytes RunningTaskCount PendingTaskCount StorageReadBytes StorageWriteBytes CPUReservation (nur mit dem Starttyp EC2) CPUUtilization (nur mit dem Starttyp EC2) MemoryReservation (Nur EC2-Starttyp) MemoryUtilization (Nur EC2-Starttyp) GPU-Reservation (nur mit dem Starttyp EC2) instance_cpu_utilization (nur EC2-Starttyp) instance_filesystem_utilization (nur EC2-Starttyp) instance_memory_utilization (nur EC2-Starttyp)

Komponententyp	Workload-Typ	Empfohlene Metriken
		instance_network_total_bytes (nur EC2-Starttyp) java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize
ECS-Service	Any	CPUUtilization MemoryUtilization CpuUtilized MemoryUtilized NetworkRxBytes NetworkTxBytes RunningTaskCount PendingTaskCount StorageReadBytes StorageWriteBytes

Komponententyp	Workload-Typ	Empfohlene Metriken
	Java-Anwendung	CPUUtilization MemoryUtilization CpuUtilized MemoryUtilized NetworkRxBytes NetworkTxBytes RunningTaskCount PendingTaskCount StorageReadBytes StorageWriteBytes java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

Komponententyp	Workload-Typ	Empfohlene Metriken
EKS-Cluster	Any	cluster_failed_node_count
		node_cpu_reserved_capacity
		node_cpu_utilization
		node_filesystem_utilization
		node_memory_reserv ed_capacity
		node_memory_utilization
		node_network_total_bytes
		pod_cpu_reserved_capacity
		pod_cpu_utilization
		pod_cpu_utilization_over_po d_limit
		pod_memory_reserve d_capacity
		pod_memory_utilization
		pod_memory_utilization_over _pod_limit
		pod_network_rx_bytes
		pod_network_tx_bytes

Komponententyp	Workload-Typ	Empfohlene Metriken
	Java-Anwendung	cluster_failed_node_count node_cpu_reserved_capacity node_cpu_utilization node_filesystem_utilization node_memory_reserved_capacity node_memory_utilization node_network_total_bytes pod_cpu_reserved_capacity pod_cpu_utilization pod_cpu_utilization_over_pod_limit pod_memory_reserved_capacity pod_memory_utilization pod_memory_utilization_over_pod_limit pod_network_rx_bytes pod_network_tx_bytes java_lang_threading_threadcount java_lang_classloading_loadedclasscount

Komponententyp	Workload-Typ	Empfohlene Metriken
		java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

Komponententyp	Workload-Typ	Empfohlene Metriken
Kubernetes-Cluster auf EC2	Any	cluster_failed_node_count node_cpu_reserved_capacity node_cpu_utilization node_filesystem_utilization node_memory_reserv ed_capacity node_memory_utilization node_network_total_bytes pod_cpu_reserved_capacity pod_cpu_utilization pod_cpu_utilization_over_po d_limit pod_memory_reserve d_capacity pod_memory_utilization pod_memory_utilization_over _pod_limit pod_network_rx_bytes pod_network_tx_bytes

Komponententyp	Workload-Typ	Empfohlene Metriken
	Java-Anwendung	cluster_failed_node_count node_cpu_reserved_capacity node_cpu_utilization node_filesystem_utilization node_memory_reserved_capacity node_memory_utilization node_network_total_bytes pod_cpu_reserved_capacity pod_cpu_utilization pod_cpu_utilization_over_pod_limit pod_memory_reserved_capacity pod_memory_utilization pod_memory_utilization_over_pod_limit pod_network_rx_bytes pod_network_tx_bytes java_lang_threading_threadcount java_lang_classloading_loadedclasscount

Komponententyp	Workload-Typ	Empfohlene Metriken
		java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

In der folgenden Tabelle sind die empfohlenen Prozesse und Prozessmetriken für jeden Komponententyp aufgeführt. CloudWatch Application Insights empfiehlt keine Prozessüberwachung für Prozesse, die nicht auf einer Instanz ausgeführt werden.

Komponententyp	Workload-Typ	Empfohlene Vorgehensweise	Empfohlene Metriken

EC2-Instance (Windows-Server)	Microsoft IIS/.NET Web Front-End	`w3wp`	`procstat cpu_usage ,`
			
			`procstat memory_rss ,`
			
			`procstat memory_vms ,`
			
			`procstat read_bytes ,`
			
			`procstat write_bytes`

	Microsoft SQL Server Database Tier	`SQLAgent`	`procstat cpu_usage ,`

Komponententyp	Workload-Typ	Empfohlene Vorgehensweise	Empfohlene Metriken
			`procstat memory_rss` , `procstat memory_vms` , `procstat read_bytes` , `procstat write_bytes`
		`sqlservr`	`procstat cpu_usage` , `procstat memory_rss` , `procstat memory_vms` , `procstat read_bytes` , `procstat write_bytes`
		`sqlwriter`	`procstat cpu_usage` , `procstat memory_rss`

Komponententyp	Workload-Typ	Empfohlene Vorgehensweise	Empfohlene Metriken
		`Reporting ServicesS ervice`	`procstat cpu_usage ,` `procstat memory_rss`

		`MsDtsServr`	`procstat cpu_usage ,`
			`procstat memory_rss ,`
			`procstat memory_vms ,`
			`procstat read_bytes ,`
			`procstat write_bytes`

		`Msmdsrv`	`procstat cpu_usage ,`
			`procstat memory_rss ,`
			`procstat memory_vms ,`
			`procstat read_bytes ,`
			`procstat write_bytes`

Komponententyp	Workload-Typ	Empfohlene Vorgehensweise	Empfohlene Metriken

	.NET workerpool/Mid-Tier	`w3wp`	`procstat cpu_usage ,`
			`procstat memory_rss ,`
			`procstat memory_vms ,`
			`procstat read_bytes ,`
			`procstat write_bytes`

	.NET Core Tier	`w3wp`	`procstat cpu_usage ,`
			`procstat memory_rss ,`
			`procstat memory_vms ,`
			`procstat read_bytes ,`
			`procstat write_bytes`

Leistungsindikator-Metriken

Leistungsindikator-Metriken werden nur für Instances empfohlen, wenn die entsprechenden Leistungsindikatorsätze auf den Windows-Instances installiert sind.

Namen von Leistungsindikator-Metriken	Name des Leistungsindikator-Metriksatzes
.NET CLR-Ausnahmenanzahl der ausgelösten Ausnahmen	.NET CLR-Ausnahmen
.NET CLR-Ausnahmenanzahl der ausgelösten Ausnahmen/Sek.	.NET CLR-Ausnahmen
.NET CLR-Ausnahmenanzahl der Filter/Sek.	.NET CLR-Ausnahmen
.NET-CLR-Ausnahmeanzahl der Finallys/Sek.	.NET CLR-Ausnahmen
.NET CLR-Ausnahmen zur Erfassung der Tiefe/Sek.	.NET CLR-Ausnahmen
.NET CLR-Interop-Anzahl der CCWs	.NET CLR Interop
.NET CLR Interop-Anzahl der Stubs	.NET CLR Interop
.NET CLR-Interop-Anzahl der TLB-Exporte/Sek.	.NET CLR Interop
.NET CLR-Interop-Anzahl der TLB-Importe/Sek.	.NET CLR Interop
.NET CLR-Interop-Anzahl Marshaling	.NET CLR Interop
.NET CLR Jit Zeitprozentsatz in Jit	.NET CLR Jit
.NET CLR Jit Standard-Jit-Fehler	.NET CLR Jit
.NET CLR Laden Zeitprozentsatz für Laden	.NET CLR Laden
.NET CLR-Laderate von Ladefehlern	.NET CLR Laden
LocksAndThreads .NET-CLR-Konfliktrate/Sekunde	.NET CLR LocksAndThreads
.NET CLR-Warteschlangenlänge/Sekunde LocksAndThreads	.NET CLR LocksAndThreads

Namen von Leistungsindikator-Metriken	Name des Leistungsindikator-Metriksatzes
.NET CLR Speicher Anzahl der insges. best. Bytes	.NET CLR-Arbeitsspeicher
.NET CLR-Arbeitsspeicherzeit in % in GC	.NET CLR-Arbeitsspeicher
.NET CLR Networking HttpWebRequest 4.0.0.0 Durchschnittliche Warteschlangenzeit	.NET CLR Networking 4.0.0.0
.NET CLR Networking 4.0.0.0 Abgebrochen/Sek HttpWebRequests	.NET CLR Networking 4.0.0.0
.NET CLR-Netzwerk 4.0.0.0 HttpWebRequests ausgefallen/Sek	.NET CLR Networking 4.0.0.0
.NET CLR-Netzwerk 4.0.0.0 in Warteschl angen/Sek HttpWebRequests	.NET CLR Networking 4.0.0.0
APP_POOL_WAS Ping-Fehler insgesamt beim Worker-Prozess	APP_POOL_WAS
ASP.NET Anwendungsneustarts	ASP.NET
ASP.NET-Anforderungen abgelehnt	ASP.NET
ASP.NET Worker-Prozess wird neu gestartet	ASP.NET
ASP.NET Anwendungs-Cache-API-Trim	ASP.NET-Anwendungen
ASP.NET-Anwendungen% verwaltete Prozessorzeit (geschätzt)	ASP.NET-Anwendungen
ASP.NET-Anwendungsfehler insgesamt/s	ASP.NET-Anwendungen
ASP.NET-Anwendungsfehler, die während der Ausführung nicht verarbeitet werden/s	ASP.NET-Anwendungen
ASP.NET-Anwendungsanforderungen in der Anwendungswarteschlange	ASP.NET-Anwendungen

Namen von Leistungsindikator-Metriken	Name des Leistungsindikator-Metriksatzes
ASP.NET-Anwendungen – Anforderungen/s	ASP.NET-Anwendungen
ASP.NET Anforderungswartezeit	ASP.NET
ASP.NET Anforderungen in Warteschlange	ASP.NET
Datenbank ==> Instanzen Datenbank Cache % Treffer	Datenbank ===> Instances
Datenbank ==> Instanzen I/O-Datenbank Lesevorgänge durchschnittliche Latenz	Datenbank ===> Instances
Datenbank ==> Instanzen I/O Database Lesevorgänge/Sek	Datenbank ===> Instances
Datenbank ==> Instanzen I/O Log Schreibvorgänge durchschnittliche Latenz	Datenbank ===> Instances
DirectoryServices Ausstehende DRA-Replikationsvorgänge	DirectoryServices
DirectoryServices Ausstehende DRA-Replikationssynchronisationen	DirectoryServices
DirectoryServices LDAP-Bindungszeit	DirectoryServices
DNS Rekursive Abfragen/Sek.	DNS
Ausfall der rekursiven DNS-Abfrage/Sek.	DNS
Empfangene DNS-TCP-Abfrage/Sek.	DNS
Gesamt empfangene DNS-Abfrage/Sek.	DNS
Gesamt gesendete DNS-Antwort/Sek.	DNS
Empfangene DNS-UDP-Abfrage/Sek.	DNS

Namen von Leistungsindikator-Metriken	Name des Leistungsindikator-Metriksatzes
Warteschlangen für HTTP-Dienstanforderungen CurrentQueueSize	HTTP-Warteschlangen für Serviceanfragen
LogicalDisk % freier Speicherplatz	LogicalDisk
LogicalDisk Durchschn. Disk sec/Write	LogicalDisk
LogicalDisk Durchschn. Festplatte Sek/Lesen	LogicalDisk
LogicalDisk Durchschn. Länge der Datenträgerwarteschlange	LogicalDisk
Speicher % übertragene Bytes im Gebrauch	Arbeitsspeicher
Verfügbarer Speicher Mbytes	Arbeitsspeicher
Speicherseiten/Sek.	Arbeitsspeicher
Langfristige durchschnittliche Standby-Cache-Lebensdauer (s) Speicher	Arbeitsspeicher
Netzwerkschnittstellen-Bytes gesamt/Sekunde	Netzwerkschnittstelle
Empfangene Netzwerkschnittstellen-Bytes/Sek.	Netzwerkschnittstelle
Gesamt gesendete Netzwerkschnittstellen-Bytes/Sek.	Netzwerkschnittstelle
Aktuelle Bandbreite der Netzwerkschnittstelle	Netzwerkschnittstelle
Auslagerungsdatei % Verwendung	Auslagerungsdatei
PhysicalDisk % Festplattenzeit	PhysicalDisk
PhysicalDisk Durchschn. Länge der Datenträgerwarteschlange	PhysicalDisk
PhysicalDisk Durchschn. Festplatte Sek./Lesen	PhysicalDisk

Namen von Leistungsindikator-Metriken	Name des Leistungsindikator-Metriksatzes
PhysicalDisk Durchschn. Disk sec/Write	PhysicalDisk
PhysicalDisk Gelesene Festplatten-Bytes/Sek	PhysicalDisk
PhysicalDisk Lesevorgänge auf der Festplatte pro Sekunde	PhysicalDisk
PhysicalDisk Bytes/Sekunde beim Schreiben auf die Festplatte	PhysicalDisk
PhysicalDisk Schreibvorgänge auf der Festplatte pro Sekunde	PhysicalDisk
Processor % Idle Time	Prozessor
Processor % Interrupt Time	Prozessor
Processor % Processor Time	Prozessor
Processor % User Time	Prozessor
SharePoint Füllgrad des festplattenbasierten Cache-Blob-Caches	SharePoint Festplattenbasierter Cache
SharePoint Festplattenbasierter Cache-Blob-Cache leert/Sekunde	SharePoint Festplattenbasierter Cache
SharePoint Trefferquote beim Blob-Cache auf festplattenbasiertem Cache	SharePoint Festplattenbasierter Cache
SharePoint Festplattenbasierter Cache Gesamtzahl der Cache-Komprimierungen	SharePoint Festplattenbasierter Cache
SharePoint Ausführungszeit/Seitenanforderung der Foundation	SharePoint Stiftung
SharePoint Cache veröffentlichen Cache wird geleert//Sekunde	SharePoint Cache veröffentlichen

Namen von Leistungsindikator-Metriken	Name des Leistungsindikator-Metriksatzes
Systemweite Sicherheitsstatistiken Kerberos-Authentifizierungen	Systemweite Sicherheitsstatistiken
Systemweite Sicherheitsstatistiken NTLM-Authentifizierungen	Systemweite Sicherheitsstatistiken
SQLServer:Access Methods Forwarded Records/Sec	SQLServer: Zugriffsmethoden
SQLServer:Access Methods Full Scans/Sec	SQLServer: Zugriffsmethoden
SQLServer:Access Methods Page Splits/Sec	SQLServer: Zugriffsmethoden
SQLServer:Buffer Manager Buffer cache hit ratio	SQLServer: Buffer Manager
SQLServer:Buffer Manager Page life expectancy	SQLServer: Buffer Manager
SQLServer:Database Replica File Bytes Received/sec	SQLServer: Datenbankreplikat
SQLServer:Database Replica Log Bytes Received/sec	SQLServer: Datenbankreplikat
SQLServer:Database Replica Log remaining for undo	SQLServer: Datenbankreplikat
SQLServer:Database Replica Log Send Queue	SQLServer: Datenbankreplikat
SQLServer:Database Replica Mirrored Write Transaction/sec	SQLServer: Datenbankreplikat
SQLServer:Database Replica Recovery Queue	SQLServer: Datenbankreplikat
SQLServer:Database Replica Redo Bytes Remaining	SQLServer: Datenbankreplikat

Namen von Leistungsindikator-Metriken	Name des Leistungsindikator-Metriksatzes
SQLServer:Database Replica Redone Bytes/sec	SQLServer: Datenbankreplikat
SQLServer:Database Replica Total Log requiring undo	SQLServer: Datenbankreplikat
SQLServer:Database Replica Transaction Delay	SQLServer: Datenbankreplikat
SQLServer:General Statistics Processes blocked	SQLServer: Allgemeine Statistiken
SQLServer:General Statistics User Connections	SQLServer: Allgemeine Statistiken
SQLServer:Latches Average Latch Wait Time (ms)	SQLServer: Latches
SQLServer:Locks Average Wait Time (ms)	SQLServer:Locks
SQLServer:Locks Lock Timeouts/Sec	SQLServer:Locks
SQLServer:Locks Lock Waits/Sec	SQLServer:Locks
SQLServer:Locks Number of Deadlocks/Sec	SQLServer:Locks
SQLServer:Memory Manager Memory Grants Pending	SQLServer: Memory Manager
SQLServer:SQL Statistics Batch Requests/sec	SQLServer: SQL-Statistiken
SQLServer:SQL Statistics SQL Compilations/Sec	SQLServer: SQL-Statistiken
SQLServer:SQL Statistics SQL Re-Compilations/Sec	SQLServer: SQL-Statistiken
System Processor Queue Length	System (System)

Namen von Leistungsindikator-Metriken	Name des Leistungsindikator-Metriksatzes
TCPv4 Connections Established	TCPv4
TCPv6 Connections Established	TCPv6
W3SVC_W3WP Dateicache-Leerungen	W3SVC_W3WP
W3SVC_W3WP Datei-Cache-Fehlschläge	W3SVC_W3WP
W3SVC_W3WP Anforderungen/Sek.	W3SVC_W3WP
W3SVC_W3WP URI-Cache-Leerungen	W3SVC_W3WP
W3SVC_W3WP URI-Cache-Fehlschläge	W3SVC_W3WP
Empfangene Web-Service-Bytes/s	Web Service
Gesendete Web-Service-Bytes/s	Web Service
Web-Service-Verbindungsversuche/Sek.	Web Service
Aktuelle Webservice-Verbindungen	Web Service
Web-Service-Abrufanforderungen/Sek.	Web Service
Webservice-Beitragsanforderungen/Sek.	Web Service

Verwenden der Ressourcenzustandsansicht in der CloudWatch Konsole

Sie können die Ressourcenintegritätsansicht verwenden, um den Zustand und die Leistung von Hosts in ihren Anwendungen automatisch in einer einzigen Ansicht zu erkennen, zu verwalten und zu visualisieren. Sie können die Integrität ihrer Hosts anhand einer Leistungs-Dimension wie CPU oder Arbeitsspeicher visualisieren und Hunderte von Hosts in einer einzigen Ansicht mithilfe von Filtern schneiden und würfeln. Sie können nach Tags oder Anwendungsfällen filtern, z. B. Hosts in derselben Auto-Scaling-Gruppe oder Hosts, die denselben Load Balancer verwenden,

Voraussetzungen

Überprüfen Sie, ob Sie die folgenden Voraussetzungen erfüllen, um sicherzustellen, dass Sie die vollen Vorteile der Ansicht Ressourcenintegrität nutzen.

- Um die Speicherauslastung Ihrer Hosts zu sehen und sie als Filter zu verwenden, müssen Sie den CloudWatch Agenten auf Ihren Hosts installieren und ihn so einrichten, dass er eine Speichermetrik CloudWatch an den `CWAgent` Standard-Namespace sendet. Auf Linux- und macOS-Instances muss der CloudWatch Agent die `mem_used_percent` Metrik senden. Auf Windows-Instanzen muss der Agent die `Memory % Committed Bytes In Use`-Metrik senden. Diese Metriken sind enthalten, wenn Sie den Assistenten verwenden, um die CloudWatch Agenten-Konfigurationsdatei zu erstellen und einen der vordefinierten Messwertsätze auszuwählen. Die vom CloudWatch Agenten gesammelten Metriken werden als benutzerdefinierte Metriken abgerechnet. Weitere Informationen finden Sie unter Den CloudWatch Agenten installieren.

  Wenn Sie den CloudWatch Agenten verwenden, um diese Speichermetriken für die Ressourcenzustandsansicht zu sammeln, müssen Sie den folgenden Abschnitt in die CloudWatch Agenten-Konfigurationsdatei aufnehmen. Dieser Abschnitt enthält die Standard-Dimensionseinstellungen und wird standardmäßig erstellt. Ändern Sie daher keinen Teil dieses Abschnitts in etwas anderes als das, was im folgenden Beispiel gezeigt wird.

  ```
  "append_dimensions": {
    "ImageId": "${aws:ImageId}",
    "InstanceId": "${aws:InstanceId}",
    "InstanceType": "${aws:InstanceType}",
    "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
  },
  ```

- Um alle Informationen in der Ansicht Ressourcenintegrität anzeigen zu können, müssen Sie an einem Konto angemeldet sein, das über die folgenden Berechtigungen verfügt. Wenn Sie mit weniger Berechtigungen angemeldet sind, können Sie weiterhin die Ressourcenintegritätsansicht verwenden, aber einige Leistungsdaten sind nicht sichtbar.

  ```
  {
      "Version": "2012-10-17",
      "Statement": [
          {
              "Action": [
                  "autoscaling:Describe*",
                  "cloudwatch:Describe*",
  ```

```
                "cloudwatch:Describe*",
                "cloudwatch:Get*",
                "cloudwatch:List*",
                "logs:Get*",
                "logs:Describe*",
                "sns:Get*",
                "sns:List*",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeRegions"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

So zeigen Sie die Ressourcenintegrität in Ihrem Konto an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Infrastrukturüberwachung, Ressourcenzustand.

   Die Seite „Ressourcenzustand“ wird angezeigt, auf der für jeden Host in Ihrem Konto ein Quadrat angezeigt wird. Jedes Quadrat wird basierend auf dem aktuellen Status dieses Hosts gefärbt, basierend auf der Einstellung für Farbe nach. Hostquadrate mit einem Alarmsymbol haben einen oder mehrere Alarme, die sich derzeit im ALARM-Status befinden.

   Sie können bis zu 500 Hosts in einer einzigen Ansicht anzeigen. Wenn Sie mehr Hosts in Ihrem Konto haben, verwenden Sie die Filtereinstellungen in Schritt 6 dieses Verfahrens.
3. Um zu ändern, welche Kriterien verwendet werden, um den Zustand jedes Hosts anzuzeigen, wählen Sie eine Einstellung für Farbe nach. Sie können CPU-Auslastung, Speicherauslastung oder Statusprüfung auswählen. Messdaten zur Speicherauslastung sind nur für Hosts verfügbar, auf denen der CloudWatch Agent ausgeführt wird und der so konfiguriert ist, dass er Speichermesswerte sammelt und an den `CWAgent` Standard-Namespace sendet. Weitere Informationen finden Sie unter Erfassen Sie mit dem CloudWatch Agenten Metriken, Logs und Traces.
4. Um die Schwellenwerte und die Farben zu ändern, die für die Zustandsindikatoren im Raster verwendet werden, wählen Sie das Zahnradsymbol über dem Raster.

5. Um umzuschalten, ob Alarme im Hostraster angezeigt werden sollen, aktivieren oder deaktivieren Sie Alarme für alle Metriken anzeigen.
6. Um die Hosts in der Karte in Gruppen aufzuteilen, wählen Sie ein Gruppierungskriterium für Gruppieren nach.
7. Um die Ansicht auf weniger Hosts einzuschränken, wählen Sie ein Filterkriterium für Filtern nach. Sie können nach Tags und Ressourcengruppierungen wie Auto-Scaling-Gruppe, Instance-Typ, Sicherheitsgruppe und mehr filtern.
8. Um Hosts zu sortieren, wählen Sie ein Sortierkriterium für Sortieren nach aus. Sie können nach Statusprüfungsergebnissen, Instance-Zustand, CPU- oder Speicherauslastung und der Anzahl der Alarme im ALARM-Zustand sortieren.
9. Um weitere Informationen zu einem Host anzuzeigen, wählen Sie das Quadrat aus, das diesen Host darstellt. Es wird ein Popup-Bereich angezeigt. Um dann tiefer in die Informationen zu diesem Host einzutauchen, wählen Sie Dashboard anzeigen oder In Liste anzeigen.

CloudWatch kontenübergreifende Beobachtbarkeit

Mit der CloudWatch kontenübergreifenden Observability von Amazon können Sie Anwendungen überwachen und Fehler beheben, die sich über mehrere Konten innerhalb einer Region erstrecken. Suchen, visualisieren und analysieren Sie nahtlos Ihre Metriken, Protokolle, Traces, Application Insights-Anwendungen und Internet Monitor-Monitore in allen verknüpften Konten ohne Kontogrenzen.

Richten Sie ein oder mehrere AWS Konten als Überwachungskonten ein und verknüpfen Sie sie mit mehreren Quellkonten. Ein Überwachungskonto ist ein zentrales AWS -Konto, das aus Quellkonten generierte Beobachtbarkeits-Daten anzeigen und mit ihnen interagieren kann. Ein Quellkonto ist ein AWS Einzelkonto, das Beobachtbarkeitsdaten für die darin enthaltenen Ressourcen generiert. Quellkonten teilen ihre Beobachtbarkeits-Daten mit dem Überwachungskonto. Die gemeinsam genutzten Beobachtbarkeits-Daten können die folgenden Arten von Telemetrie umfassen:

- Metriken bei Amazon CloudWatch. Sie können wählen, ob Sie die Metriken aus allen Namespaces mit dem Monitoring-Konto teilen oder nach einer Teilmenge von Namespaces filtern möchten.
- Gruppen in Amazon CloudWatch Logs protokollieren. Sie können wählen, ob Sie alle Protokollgruppen mit dem Überwachungskonto teilen oder nach einer Teilmenge von Protokollgruppen filtern möchten.
- Spuren in AWS X-Ray
- Anwendungen in Amazon CloudWatch Application Insights
- Monitore im CloudWatch Internetmonitor

Um Verknüpfungen zwischen Überwachungskonten und Quellkonten herzustellen, können Sie die CloudWatch Konsole verwenden. Verwenden Sie alternativ die Observability Access Manager-Befehle in der API AWS CLI und. Weitere Informationen finden Sie unter Observability-Access-Manager-API-Referenz.

Eine Senke ist eine Ressource, die einen Zuordnungspunkt in einem Überwachungskonto darstellt. Quellkonten können eine Verknüpfung zur Senke herstellen, um Beobachtbarkeits-Daten auszutauschen. Jedes Konto kann eine Senke pro Region haben. Jede Senke wird von dem Überwachungskonto verwaltet, in dem sie sich befindet. Ein Beobachtbarkeits-Link ist eine Ressource, die die Verbindung darstellt, die zwischen einem Quellkonto und einem Überwachungskonto hergestellt wurde. Links werden vom Quellkonto verwaltet.

Eine Videodemonstration der Einrichtung CloudWatch kontenübergreifender Observability finden Sie im folgenden Video.

Im nächsten Thema wird erklärt, wie die CloudWatch kontenübergreifende Beobachtbarkeit sowohl für Begleitkonten als auch für Quellkonten eingerichtet wird. Informationen zum kontenübergreifenden Dashboard für regionsübergreifende CloudWatch Konten finden Sie unter. Kontoübergreifende, regionsübergreifende Konsole CloudWatch

Organisationen für Quellkonten verwenden

Es gibt zwei Möglichkeiten, Quellkonten mit Ihrem Überwachungskonto zu verknüpfen. Sie können eine oder beide Optionen verwenden.

- Wird verwendet AWS Organizations , um Konten in einer Organisation oder Organisationseinheit mit dem Überwachungskonto zu verknüpfen.
- Connect einzelne AWS Konten mit dem Überwachungskonto.

Wir empfehlen Ihnen, Organizations zu verwenden, damit neue AWS Konten, die später in der Organisation erstellt werden, automatisch als Quellkonten in die kontenübergreifende Observability aufgenommen werden.

Details zur Verknüpfung von Überwachungskonten und Quellkonten

- Jedes Überwachungskonto kann mit bis zu 100 000 Quellkonten verknüpft werden.
- Jedes Quellkonto kann Daten mit bis zu fünf Überwachungskonten teilen.
- Sie können ein einzelnes Konto sowohl als Überwachungskonto als auch als Quellkonto einrichten. Wenn Sie dies tun, sendet dieses Konto nur die Beobachtbarkeits-Daten von sich selbst an das verknüpfte Überwachungskonto. Es leitet die Daten von seinen Quellkonten nicht weiter.
- Ein Überwachungskonto gibt an, welche Telemetrietypen mit ihm gemeinsam genutzt werden können. Ein Quellkonto gibt an, welche Telemetrietypen es teilen möchte.
  - Wenn im Überwachungskonto mehr Telemetrietypen ausgewählt sind als im Quellkonto, werden die Konten verknüpft. Nur die Datentypen, die in beiden Konten ausgewählt wurden, werden gemeinsam genutzt.
  - Wenn im Quellkonto mehr Telemetrietypen ausgewählt sind als im Überwachungskonto, schlägt die Erstellung der Verknüpfung fehl und es wird nichts freigegeben.

  - Ein Metrikname wird erst in der Monitoring-Kontokonsole angezeigt, wenn diese Metrik nach der Erstellung des Links neue Datenpunkte ausgibt.
- Um eine Verknüpfung zwischen Konten zu entfernen, tun Sie dies vom Quellkonto aus.
- Um eine Senke in einem Monitoring-Konto zu löschen, müssen Sie zuerst alle Links zu dieser Senke mit dem Monitoring-Konto entfernen.

Preise

CloudWatch Bei kontenübergreifender Observability fallen keine zusätzlichen Kosten für Logs und Metriken an, und die erste Trace-Kopie ist kostenlos. Weitere Informationen zur Preisgestaltung finden Sie unter CloudWatchAmazon-Preise.

Inhalt

Überwachungskonten mit Quellkonten verknüpfen

Die Themen in diesem Abschnitt beschreiben, wie Sie Verknüpfungen zwischen Überwachungskonten und Quellkonten einrichten.

Wir empfehlen Ihnen, ein neues AWS Konto zu erstellen, das als Überwachungskonto für Ihre Organisation dient.

Inhalt

Erforderliche Berechtigungen

Um Links zwischen einem Überwachungskonto und einem Quellkonto zu erstellen, müssen Sie mit bestimmten Berechtigungen angemeldet sein.

- Um ein Überwachungskonto einzurichten – Sie müssen entweder vollen Administratorzugriff auf das Überwachungskonto haben, oder Sie müssen sich mit den folgenden Berechtigungen bei diesem Konto anmelden:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSinkModification",
            "Effect": "Allow",
            "Action": [
                "oam:CreateSink",
                "oam:DeleteSink",
                "oam:PutSinkPolicy",
                "oam:TagResource"
            ],
            "Resource": "*"
        },
        {
```

```
            "Sid": "AllowReadOnly",
            "Effect": "Allow",
            "Action": ["oam:Get*", "oam:List*"],
            "Resource": "*"
        }
    ]
}
```

- Quellkonto, auf ein bestimmtes Überwachungskonto beschränkt – Um Links für nur ein bestimmtes Überwachungskonto zu erstellen, zu aktualisieren und zu verwalten, müssen Sie sich mit mindestens den folgenden Berechtigungen bei einem Konto anmelden. In diesem Beispiel ist das Überwachungskonto 999999999999.

  Wenn der Link nicht alle fünf Ressourcentypen (Metriken, Protokolle, Traces, Application Insights-Anwendungen und Internet Monitor-Monitore) gemeinsam nutzen soll, können Sie`cloudwatch:Link`, `logs:Link` `xray:Link``applicationinsights:Link`, oder nach `internetmonitor:Link` Bedarf weglassen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink",
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
            ],
            "Effect": "Allow",
            "Resource": "arn:*:oam:*:*:link/*"
        },
        {
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Effect": "Allow",
            "Resource": "arn:*:oam:*:*:sink/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": [
```

```
                        "999999999999"
                    ]
                }
            }
        },
        {
            "Action": "oam:ListLinks",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "cloudwatch:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "logs:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "xray:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
             "Action": "applicationinsights:Link",
             "Effect": "Allow",
             "Resource": "*"
         },
        {
             "Action": "internetmonitor:Link",
             "Effect": "Allow",
             "Resource": "*"
        }
    ]
}
```

- Quellkonto mit Berechtigungen zum Verknüpfen mit einem beliebigen Überwachungskonto — Um einen Link zu einem vorhandenen Überwachungskonto zu erstellen und Metriken, Protokollgruppen, Traces, Application Insights-Anwendungen und Internet Monitor-Monitore gemeinsam zu nutzen, müssen Sie sich mit vollen Administratorrechten beim Quellkonto anmelden oder sich dort mit den folgenden Berechtigungen anmelden

Wenn der Link nicht alle fünf Ressourcentypen (Metriken, Protokolle, Traces, Application Insights-Anwendungen und Internet Monitor-Monitore) gemeinsam nutzen soll, können Sie`cloudwatch:Link`,, `logs:Link` `xray:Link``applicationinsights:Link`, oder nach `internetmonitor:Link` Bedarf weglassen.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Resource": [
                "arn:aws:oam:*:*:link/*",
                "arn:aws:oam:*:*:sink/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:List*",
                "oam:Get*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
            ],
            "Resource": "arn:aws:oam:*:*:link/*"
        },
        {
            "Action": "cloudwatch:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "xray:Link",
```

```
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "logs:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
             "Action": "applicationinsights:Link",
             "Effect": "Allow",
             "Resource": "*"
        },
        {
             "Action": "internetmonitor:Link",
             "Effect": "Allow",
             "Resource": "*"
        }
    ]
}
```

Übersicht über die Einrichtung

Die folgenden allgemeinen Schritte zeigen Ihnen, wie Sie CloudWatch kontenübergreifende Observability einrichten.

> **Note**
>
> Wir empfehlen, ein neues AWS Konto zu erstellen, das als Überwachungskonto Ihrer Organisation verwendet werden kann.

1. Richten Sie einen dedizierten Überwachungskonto ein.
2. (Optional) Laden Sie eine AWS CloudFormation Vorlage herunter oder kopieren Sie eine URL, um Quellkonten zu verknüpfen.
3. Verknüpfen Sie Quellkonten mit dem Überwachungskonto.

Nachdem Sie diese Schritte abgeschlossen haben, können Sie das Überwachungskonto verwenden, um die Beobachtbarkeits-Daten der Quellkonten einzusehen.

Schritt 1: Einrichten eines Überwachungskontos

Folgen Sie den Schritten in diesem Abschnitt, um ein AWS Konto als Überwachungskonto für CloudWatch kontenübergreifende Beobachtbarkeit einzurichten.

Voraussetzungen

- Wenn Sie Konten in einer AWS Organizations Organisation als Quellkonten einrichten, rufen Sie den Organisationspfad oder die Organisations-ID ab.
- Wenn Sie für die Quellkonten nicht Organizations verwenden – Rufen Sie die Konto-IDs der Quellkonten ab.

Um ein Konto als Überwachungskonto einzurichten, müssen Sie über bestimmte Berechtigungen verfügen. Weitere Informationen finden Sie unter Erforderliche Berechtigungen.

So richten Sie ein Überwachungskonto ein

1. Melden Sie sich bei dem Konto an, das Sie als Überwachungskonto verwenden möchten.
2. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
3. Wählen Sie im linken Navigationsbereich die Option Einstellungen aus.
4. Wählen Sie unter Monitoring account configuration (Konfiguration des Überwachungskontos) die Option Configure (Konfigurieren) aus.
5. Wählen Sie für Select data aus, ob dieses Monitoring-Konto Logs, Metrics, Traces, Application Insights — Applications und Internet Monitor — Monitoring-Daten von den Quellkonten anzeigen kann, mit denen es verknüpft ist.
6. Geben Sie unter List source accounts (Quellkonten auflisten) die Quellkonten ein, die dieses Überwachungskonto anzeigen soll. Um die Quellkonten zu identifizieren, geben Sie individuelle Konto-IDs, Organisationspfade oder Organisations-IDs ein. Wenn Sie einen Organisationspfad oder eine Organisations-ID eingeben, darf dieses Überwachungskonto Beobachtbarkeits-Daten aus allen verknüpften Konten in dieser Organisation anzeigen.

   Trennen Sie die Einträge in der Liste durch Kommas.

> **Important**
>
> Wenn Sie einen Organisationspfad eingeben, halten Sie sich an das genaue Format. Die OU-ID muss mit einem / (einem Schrägstrich) enden. Beispiel: `o-a1b2c3d4e5/r-f6g7h8i9j0example/ou-def0-awsbbbb/`

7. Geben Sie unter Define a label to identify your source account (Ein Label zur Identifizierung Ihres Quellkontos definieren) an, ob Kontonamen oder E-Mail-Adressen verwendet werden sollen, um die Quellkonten zu identifizieren, wenn Sie das Überwachungskonto verwenden, um sie anzuzeigen.
8. Wählen Sie Konfigurieren aus.

> **Important**
>
> Die Verknüpfung zwischen dem Überwachungs- und dem Quellkonto ist erst abgeschlossen, wenn Sie die Quellkonten konfiguriert haben. Weitere Informationen finden Sie in den folgenden Abschnitten.

Schritt 2: (Optional) Laden Sie eine AWS CloudFormation Vorlage oder URL herunter

Um Quellkonten mit einem Überwachungskonto zu verknüpfen, empfehlen wir die Verwendung einer AWS CloudFormation -Vorlage oder einer URL.

- Wenn Sie eine gesamte Organisation verknüpfen, wird CloudWatch eine AWS CloudFormation Vorlage bereitgestellt.
- Wenn Sie einzelne Konten verknüpfen, verwenden Sie entweder eine AWS CloudFormation Vorlage oder eine URL, die Folgendes CloudWatch bereitstellt.

Um eine AWS CloudFormation Vorlage zu verwenden, müssen Sie sie während dieser Schritte herunterladen. Nachdem Sie das Überwachungskonto mit mindestens einem Quellkonto verknüpft haben, steht die AWS CloudFormation Vorlage nicht mehr zum Herunterladen zur Verfügung.

Um eine AWS CloudFormation Vorlage herunterzuladen oder eine URL für die Verknüpfung von Quellkonten mit dem Überwachungskonto zu kopieren

1. Melden Sie sich bei dem Konto an, das Sie als Überwachungskonto verwenden möchten.
2. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
3. Wählen Sie im linken Navigationsbereich die Option Einstellungen aus.
4. Wählen Sie unter Monitoring account configuration (Konfiguration des Überwachungskontos) die Option Resources to link accounts (Ressourcen, um Konten zu verknüpfen) aus.
5. Führen Sie eine der folgenden Aktionen aus:

   - Wählen Sie AWS -Organisation aus, um eine Vorlage zu erhalten, mit der Sie Konten in einer Organisation mit diesem Überwachungskonto verknüpfen können.
   - Wählen Sie Any account (Beliebiges Konto), um eine Vorlage oder URL für die Einrichtung einzelner Konten als Quellkonten zu erhalten.
6. Führen Sie eine der folgenden Aktionen aus:

   - Wenn Sie AWS Organisation ausgewählt haben, wählen Sie CloudFormation Vorlage herunterladen.
   - Wenn Sie Beliebiges Konto ausgewählt haben, wählen Sie entweder CloudFormation Vorlage herunterladen oder URL kopieren.
7. (Optional) Wiederholen Sie die Schritte 5 bis 6, um sowohl die AWS CloudFormation Vorlage als auch die URL herunterzuladen.

Schritt 3: Die Quellkonten verknüpfen

Führen Sie die Schritte in diesen Abschnitten aus, um Quellkonten mit einem Überwachungskonto zu verknüpfen.

Um Überwachungskonten mit Quellkonten zu verknüpfen, müssen Sie über bestimmte Berechtigungen verfügen. Weitere Informationen finden Sie unter Erforderliche Berechtigungen.

Verwenden Sie eine AWS CloudFormation -Vorlage, um alle Konten in einer Organisation oder einer Organisationseinheit als Quellkonten einzurichten

Bei diesen Schritten wird davon ausgegangen, dass Sie die erforderliche AWS CloudFormation Vorlage bereits heruntergeladen haben, indem Sie die unter beschriebenen Schritte ausführenSchritt 2: (Optional) Laden Sie eine AWS CloudFormation Vorlage oder URL herunter.

So verwenden Sie eine AWS CloudFormation Vorlage, um Konten in einer Organisation oder Organisationseinheit mit dem Überwachungskonto zu verknüpfen

1. Melden Sie sich beim Verwaltungskonto Ihrer Organisation an.
2. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
3. Wählen Sie in der linken Navigationsleiste StackSets.
4. Vergewissern Sie sich, dass Sie in der gewünschten Region angemeldet sind, und wählen Sie dann Erstellen aus StackSet.
5. Wählen Sie Weiter aus.
6. Wählen Sie Template is ready (Vorlage ist bereit) und wählen Sie Upload a template file (Eine Vorlagendatei hochladen).
7. Wählen Sie Choose file (Datei auswählen), wählen Sie die Vorlage aus, die Sie vom Überwachungskonto heruntergeladen haben, und wählen Sie Open (Öffnen).
8. Wählen Sie Weiter aus.
9. Geben Sie unter StackSet Details angeben einen Namen für die ein StackSet und klicken Sie auf Weiter.
10. Wählen Sie für Add stacks to stack set (Stacks zum Stack-Set hinzufügen) Deploy new stacks (Neue Stacks bereitstellen).
11. Wählen Sie unter Deployment targets (Bereitstellungsziele) aus, ob die Bereitstellung für die gesamte Organisation oder für bestimmte Organisationseinheiten erfolgen soll.
12. Wählen Sie unter Regionen angeben aus, in welchen Regionen CloudWatch kontenübergreifende Observability bereitgestellt werden soll.
13. Wählen Sie Weiter aus.
14. Überprüfen Sie auf der Seite Review (Überprüfen) Ihre Optionen und wählen Sie anschließend Submit (Übermitteln) aus.
15. Aktualisieren Sie auf der Registerkarte Stack-Instances den Bildschirm, bis Sie sehen, dass Ihre Stack-Instances den Status CREATE_COMPLETE (ERSTELLUNG_KOMPLETT) haben.

Eine AWS CloudFormation -Vorlage verwenden, um individuelle Quellkonten einzurichten

Bei diesen Schritten wird davon ausgegangen, dass Sie die erforderliche AWS CloudFormation Vorlage bereits heruntergeladen haben, indem Sie die unter beschriebenen Schritte ausführen. Schritt 2: (Optional) Laden Sie eine AWS CloudFormation Vorlage oder URL herunter

Um eine AWS CloudFormation Vorlage zur Einrichtung einzelner Quellkonten für CloudWatch kontenübergreifende Beobachtbarkeit zu verwenden

1. Melden Sie sich beim Quellkonto an.
2. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
3. Wählen Sie im linken Navigationsbereich Stacks (Stacks) aus.
4. Vergewissern Sie sich, dass Sie in der gewünschten Region angemeldet sind, und wählen Sie dann Create StackSet (Stack-Set erstellen), With new resources (standard) (Mit neuen Ressourcen (Standard)).
5. Wählen Sie Weiter aus.
6. Wählen Sie Upload a template file (Vorlagendatei hochladen).
7. Wählen Sie Choose file (Datei auswählen), wählen Sie die Vorlage aus, die Sie vom Überwchungskonto heruntergeladen haben, und wählen Sie Open (Öffnen).
8. Wählen Sie Weiter aus.
9. Geben Sie für Specify Stack details (Stack-Details angeben) einen Stack-Namen ein und wählen Sie Next (Weiter).
10. Wählen Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) Next (Weiter) aus.
11. Klicken Sie auf der Seite Review (Überprüfen) auf Submit (Übermitteln).
12. Aktualisieren Sie auf der Statusseite Ihres Stacks den Bildschirm, bis Sie sehen, dass Ihr Stapel den Status CREATE_COMPLETE (ERSTELLUNG_KOMPLETT) hat.
13. Um dieselbe Vorlage zu verwenden, um weitere Quellkonten mit diesem Überwachungskonto zu verknüpfen, melden Sie sich von diesem Konto ab und melden Sie sich beim nächsten Quellkonto an. Wiederholen Sie dann die Schritte 2-12.

Eine URL verwenden, um individuelle Quellkonten einzurichten

Bei diesen Schritten wird vorausgesetzt, dass Sie die erforderliche URL bereits heruntergeladen haben, indem Sie die Schritte unter Schritt 2: (Optional) Laden Sie eine AWS CloudFormation Vorlage oder URL herunter ausführten.

Wie Sie eine URL zu verwenden, um einzelne Quellkonten mit dem Überwachungskonto zu verknüpfen

1. Melden Sie sich bei dem Konto an, das Sie als Quellkonto verwenden möchten.
2. Geben Sie die URL ein, die Sie vom Überwachungskonto kopiert haben.

   Sie sehen die CloudWatch Einstellungsseite, auf der einige Informationen ausgefüllt sind.
3. Wählen Sie unter Daten auswählen aus, ob dieses Quellkonto Logs, Metrics, Traces, Application Insights — Anwendungen und Internet Monitor — Monitoring-Daten an dieses Überwachungskonto weitergibt.

   Sowohl für Logs als auch für Metrics können Sie wählen, ob Sie alle Ressourcen oder nur einen Teil davon mit dem Monitoring-Konto teilen möchten.

   a. (Optional) Um eine Teilmenge der Protokollgruppen dieses Kontos für das Überwachungskonto freizugeben, wählen Sie Protokolle und anschließend Protokolle filtern aus. Verwenden Sie dann das Feld „Protokolle filtern“, um eine Abfrage zu erstellen, um die Protokollgruppen zu finden, die Sie gemeinsam nutzen möchten. Die Abfrage verwendet den Begriff `LogGroupName` und einen oder mehrere der folgenden Operanden.

      - = und `!=`
      - `AND`
      - `OR`
      - ^steht für LIKE und `!^` steht für NOT LIKE. Diese können nur als Präfixsuchen verwendet werden. Fügen Sie `%` am Ende der Zeichenfolge, nach der Sie suchen und die Sie einschließen möchten, ein ein ein.
      - `IN`und `NOT IN` mithilfe von Klammern () ( )

      Die vollständige Abfrage darf nicht mehr als 2000 Zeichen lang sein und ist auf fünf bedingte Operanden beschränkt. Bedingte Operanden sind `AND` und. `OR` Die Anzahl der anderen Operanden ist nicht begrenzt.

> **Tip**
>
> Wählen Sie Beispielabfragen anzeigen, um die richtige Syntax für gängige Abfrageformate zu sehen.

b. (Optional) Um eine Teilmenge der Metrik-Namespaces dieses Kontos mit dem Überwachungskonto gemeinsam zu nutzen, wählen Sie Metriken und dann Metriken filtern aus. Verwenden Sie dann das Feld Metriken filtern, um eine Abfrage zu erstellen, um die Metrik-Namespaces zu finden, die Sie gemeinsam nutzen möchten. Verwenden Sie den Begriff `Namespace` und einen oder mehrere der folgenden Operanden.

- = und `!=`
- `AND`
- `OR`
- `LIKE` und `NOT LIKE`. Diese können nur als Präfixsuchen verwendet werden. Fügen Sie `%` am Ende der Zeichenfolge, nach der Sie suchen und die Sie einschließen möchten, ein ein.
- `IN`und `NOT IN` mithilfe von Klammern () ( )

Die vollständige Abfrage darf nicht mehr als 2000 Zeichen lang sein und ist auf fünf bedingte Operanden beschränkt. Bedingte Operanden sind `AND` und. `OR` Die Anzahl der anderen Operanden ist nicht begrenzt.

> **Tip**
>
> Wählen Sie Beispielabfragen anzeigen, um die richtige Syntax für gängige Abfrageformate zu sehen.

4. Ändern Sie den ARN nicht unter Enter monitoring account configuration ARN (ARN für die Konfiguration des Überwachungskontos eingeben) ein.
5. Der Abschnitt Define a label to identify your source account (Ein Label zur Identifizierung Ihres Quellkontos definieren) ist mit der Labelauswahl aus dem Überwachungskonto vorausgefüllt. Wählen Sie Edit (Bearbeiten) aus, um eine Option zu ändern.
6. Wählen Sie Verknüpfen.

7. Geben Sie in das Bestätigungsfeld **`Confirm`** ein und wählen Sie dann Confirm (Bestätigen).
8. Um dieselbe URL zu verwenden, um weitere Quellkonten mit diesem Überwachungskonto zu verknüpfen, melden Sie sich von diesem Konto ab und melden Sie sich beim nächsten Quellkonto an. Wiederholen Sie dann die Schritte 2-7.

Überwachungskonten und Quellkonten verwalten

Nachdem Sie Ihre Überwachungs- und Quellkonten eingerichtet haben, können Sie die Schritte in diesen Abschnitten verwenden, um sie zu verwalten.

Inhalt

Quellkonten mit einem vorhandenen Überwachungskonto verknüpfen

Führen Sie die Schritte in diesem Abschnitt aus, um Links von zusätzlichen Quellkonten zu einem vorhandenen Überwachungskonto hinzuzufügen.

Jedes Quellkonto kann mit bis zu fünf Überwachungskonten verknüpft werden. Jedes Überwachungskonto kann mit bis zu 100 000 Quellkonten verknüpft werden.

Um ein Quellkonto zu verwalten, müssen Sie über bestimmte Berechtigungen verfügen. Weitere Informationen finden Sie unter Erforderliche Berechtigungen.

Wie Sie weitere Quellkonten mit einem vorhandenen Überwachungskonto verknüpfen

1. Melden Sie sich beim Überwachungskonto an.
2. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
3. Wählen Sie im linken Navigationsbereich die Option Einstellungen aus.
4. Wählen Sie unter Monitoring account configuration (Konfiguration des Überwachungskontos) die Option Manage source accounts (Quellkonten verwalten) aus.
5. Wählen Sie die Registerkarte Configuration policy (Konfigurationsrichtlinie) aus.
6. Fügen Sie im Feld Configuration policy (Konfigurationsrichtlinie) die neue Quellkonto-ID in der Zeile Principal (Prinzipal) hinzu.

Angenommen, die Zeile Principal (Prinzipal) sieht derzeit wie folgt aus:

```
"Principal": {"AWS": ["111111111111", "222222222222"]}
```

Um 999999999999 als drittes Quellkonto hinzuzufügen, bearbeiten Sie die Zeile wie folgt:

```
"Principal": {"AWS": ["111111111111", "222222222222", "999999999999"]}
```

7. Wählen Sie Aktualisieren.
8. Wählen Sie die Registerkarte Configuration details (Konfigurationsdetails) aus.
9. Wählen Sie das Kopiersymbol, das sich neben dem Sink-ARN des Überwachungskontos befindet.
10. Melden Sie sich bei dem Konto an, das Sie als neues Quellkonto verwenden möchten.
11. Fügen Sie den Senken-ARN des Überwachungskontos ein, den Sie in Schritt 9 kopiert haben.

    Sie sehen die CloudWatch Einstellungsseite mit einigen eingegebenen Informationen.
12. Wählen Sie für Daten auswählen, ob dieses Quellkonto die Daten von Protokollen, Metriken, Ablaufverfolgungen und Application-Insights-Anwendungen an Quellkonten sendet, mit denen es verknüpft ist.
13. Ändern Sie den ARN nicht unter Enter monitoring account configuration ARN (ARN für die Konfiguration des Überwachungskontos eingeben) ein.
14. Der Abschnitt Define a label to identify your source account (Ein Label zur Identifizierung Ihres Quellkontos definieren) ist mit der Labelauswahl aus dem Überwachungskonto vorausgefüllt. Wählen Sie Edit (Bearbeiten) aus, um eine Option zu ändern.
15. Wählen Sie Verknüpfen.
16. Geben Sie in das Bestätigungsfeld **`Confirm`** ein und wählen Sie dann Confirm (Bestätigen).

Die Verknüpfung zwischen einem Überwachungskonto und einem Quellkonto entfernen

Führen Sie die Schritte in diesem Abschnitt aus, um das Senden von Daten von einem Quellkonto an ein Überwachungskonto zu beenden.

Sie müssen über die für die Verwaltung eines Quellkontos erforderlichen Berechtigungen verfügen, um diese Aufgabe ausführen zu können. Weitere Informationen finden Sie unter Erforderliche Berechtigungen.

So entfernen Sie die Verknüpfung zwischen einem Quellkonto und einem Überwachungskonto

1. Melden Sie sich beim Quellkonto an.
2. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
3. Wählen Sie im linken Navigationsbereich die Option Einstellungen aus.
4. Wählen Sie unter Source account information (Quellkontoinformationen) die Option View monitoring accounts (Überwachungskonten anzeigen) aus.
5. Aktivieren Sie das Kontrollkästchen neben dem Überwachungskonto, mit dem Sie keine Daten mehr teilen möchten.
6. Wählen Sie Stop sharing data (Daten nicht mehr teilen), Confirm (Bestätigen).
7. Melden Sie sich beim Überwachungskonto an.
8. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
9. Wählen Sie Settings (Einstellungen) aus.
10. Wählen Sie unter Monitoring account information (Informationen zum Überwachungskonto) die Option View configuration (Konfiguration anzeigen) aus.
11. Löschen Sie im Feld Policy (Richtlinie) die Quellkonto-ID aus der Zeile Principal (Prinzipal) und wählen Sie Update (Aktualisieren).

Informationen über ein Überwachungskonto anzeigen

Führen Sie die Schritte in diesem Abschnitt aus, um die kontoübergreifenden Einstellungen eines Überwachungskontos anzuzeigen.

Um ein Überwachungskonto zu verwalten, müssen Sie über bestimmte Berechtigungen verfügen. Weitere Informationen finden Sie unter Erforderliche Berechtigungen.

So verwalten Sie ein Überwachungskonto

1. Melden Sie sich beim Überwachungskonto an.
2. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
3. Wählen Sie im linken Navigationsbereich die Option Einstellungen aus.

4. Wählen Sie unter Monitoring account configuration (Konfiguration des Überwachungskontos) die Option Manage source accounts (Quellkonten verwalten) aus.
5. Um die Beobachtbarkeits-Zugriffsmanager-Richtlinie anzuzeigen, die dieses Konto als Überwachungskonto ermöglicht, wählen Sie die Registerkarte Configuration policy (Konfigurationsrichtlinie) aus.
6. Um die Quellkonten anzuzeigen, die mit diesem Überwachungskonto verknüpft sind, wählen Sie die Registerkarte Linked source accounts (Verknüpfte Quellkonten).
7. Wählen Sie die Registerkarte Verknüpfte Quellkonten, um den ARN des Überwachungskontos und die Datentypen anzuzeigen, die dieses Überwachungskonto in der Registerkarte Linked source accounts (verknüpfte Quellkonten) anzeigen kann.

Metriken aus anderen Datenquellen abfragen

Sie können CloudWatch damit Messwerte aus anderen Datenquellen abfragen, visualisieren und Alarme erstellen. Dazu stellen Sie eine Verbindung CloudWatch zu den anderen Datenquellen her. Auf diese Weise erhalten Sie eine einzige, konsolidierte Überwachungserfahrung innerhalb der CloudWatch Konsole. Unabhängig davon, wo die Daten gespeichert sind, erhalten Sie einen einheitlichen Überblick über Ihre Infrastruktur- und Anwendungsmetriken, sodass Sie Probleme schneller identifizieren und lösen können.

Nachdem Sie mithilfe eines CloudWatch Assistenten eine Verbindung zu einer Datenquelle hergestellt haben, CloudWatch wird ein AWS CloudFormation Stack erstellt, der eine AWS Lambda Funktion bereitstellt und konfiguriert. Diese Lambda-Funktion wird bei jeder Abfrage der Datenquelle bei Bedarf ausgeführt. Der CloudWatch Abfrage-Generator zeigt Ihnen in Echtzeit eine Liste von Elementen, die abgefragt werden können, z. B. Metriken, Tabellen, Felder oder Beschriftungen. Während Sie eine Auswahl treffen, füllt der Abfragegenerator eine Abfrage vorab in der Sprache der ausgewählten Quelle aus.

CloudWatch bietet geführte Assistenten, mit denen Sie eine Verbindung zu den folgenden Datenquellen herstellen können. Für diese Datenquellen stellen Sie grundlegende Informationen zur Identifizierung der Datenquelle und der Anmeldeinformationen bereit. Sie können Konnektoren zu anderen Datenquellen auch manuell erstellen, indem Sie Ihre eigenen Lambda-Funktionen erstellen.

- Amazon OpenSearch Service — Leiten Sie Metriken aus Ihren OpenSearch Service-Logs und Traces ab.
- Amazon Managed Service für Prometheus – Fragen Sie diese Metriken mit PromQL ab.
- Amazon RDS für MySQL – Verwenden Sie SQL, um die in Ihren Amazon-RDS-Tabellen gespeicherte Daten in Metriken umzuwandeln.
- Amazon RDS für PostgreSQL – Verwenden Sie SQL, um die in Ihren Amazon-RDS-Tabellen gespeicherte Daten in Metriken umzuwandeln.
- Amazon-S3-CSV-Dateien – Zeigt Metrikdaten aus einer CSV-Datei an, die in einem Amazon-S3-Bucket gespeichert ist.
- Microsoft Azure Monitor – Fragen Sie Metriken von Ihrem Microsoft-Azure-Monitor-Konto ab.
- Prometheus – Fragen Sie diese Metriken mit PromQL ab.

Nachdem Sie Konnektoren zu Datenquellen erstellt haben, finden Sie weitere Informationen zur grafischen Darstellung einer Metrik aus einer Datenquelle unter Erstellen eines Diagramms mit

Metriken aus einer anderen Datenquelle. Informationen zum Einstellen eines Alarms für eine Metrik aus einer Datenquelle finden Sie unter Einen Alarm basierend auf einer verbundenen Datenquelle erstellen.

Themen

Verwalten des Zugriffs auf Datenquellen

CloudWatch verwendet AWS CloudFormation , um die erforderlichen Ressourcen in Ihrem Konto zu erstellen. Wir empfehlen, dass Sie die `cloudformation:TemplateUrl` Bedingung verwenden, um den Zugriff auf AWS CloudFormation Vorlagen zu kontrollieren, wenn Sie IAM-Benutzern `CreateStack` Berechtigungen erteilen.

> **Warning**
>
> Jeder Benutzer, dem Sie die Berechtigung zum Aufrufen von Datenquellen erteilen, kann Metriken aus dieser Datenquelle abfragen, auch wenn dieser Benutzer keine direkten IAM-Berechtigungen für die Datenquelle besitzt. Wenn Sie beispielsweise einem Benutzer `lambda:InvokeFunction`-Berechtigungen für eine Lambda-Funktion der Datenquelle Amazon Managed Service für Prometheus gewähren, kann dieser Benutzer Metriken aus dem entsprechenden Workspace von Amazon Managed Service für Prometheus abfragen, auch wenn Sie ihm keinen direkten IAM-Zugriff auf diesen Workspace gewährt haben.

Vorlagen-URLs für Datenquellen finden Sie auf der Seite Stack erstellen in der CloudWatch Einstellungskonsole.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect" : "Allow",
```

```
      "Action" : [ "cloudformation:CreateStack" ],
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "cloudformation:TemplateUrl" : [ data-source-template-url ]
        }
      }
    }
  ]
}
```

Weitere Informationen zur AWS CloudFormation Zugriffskontrolle finden Sie unter Zugriffskontrolle mit AWS Identity and Access Management

Mit einem Assistenten eine Verbindung zu einer vordefinierten Datenquelle herstellen

Dieses Thema enthält Anweisungen zur Verwendung des Assistenten, um eine Verbindung CloudWatch zu den folgenden Datenquellen herzustellen.

- OpenSearch Amazon-Dienst
- Amazon Managed Service für Prometheus
- Amazon RDS für MySQL
- Amazon RDS für PostgreSQL
- Amazon-S3-CSV-Dateien
- Microsoft Azure Monitor
- Prometheus

Später in diesem Abschnitt finden Sie Unterabschnitte mit Hinweisen zur Verwaltung und Abfrage jeder dieser Datenquellen.

So erstellen Sie einen Konnektor zu Datenquellen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Settings (Einstellungen).
3. Wählen Sie die Registerkarte Metrik-Datenquellen.

4. Klicken Sie auf Create data source.
5. Wählen Sie die gewünschte Quelle aus und wählen Sie dann Weiter.
6. Geben Sie einen Namen für die Datenquelle ein.
7. Geben Sie je nach der ausgewählten Datenquelle die anderen erforderlichen Informationen ein. Dies kann Anmeldeinformationen für den Zugriff auf die Datenquelle und identifizierende Informationen wie den Namen des Prometheus-Workspace, den Datenbanknamen oder den Amazon-S3-Bucket-Namen beinhalten. Bei AWS Diensten erkennt der Assistent die Ressourcen und fügt sie in das Auswahl-Dropdown-Menü ein.

   Weitere Hinweise zu der von Ihnen verwendeten Datenquelle finden Sie in den Abschnitten nach diesem Verfahren.
8. Um eine CloudWatch Verbindung mit der Datenquelle in einer VPC herzustellen, wählen Sie Use a VPC und wählen Sie die zu verwendende VPC aus. Wählen Sie danach das Subnetz und die Sicherheitsgruppe aus.
9. Wählen Sie Ich bestätige, dass AWS CloudFormation IAM-Ressourcen erstellt werden. Diese Ressource ist die Ausführungsrolle der Lambda-Funktion.
10. Klicken Sie auf Create data source.

    Die neue Quelle, die Sie gerade hinzugefügt haben, wird erst angezeigt, wenn der AWS CloudFormation Stapel sie erstellt hat. Um den Fortschritt zu überprüfen, können Sie „Status meines CloudFormation Stacks anzeigen“ wählen. Oder Sie können das Aktualisierungssymbol wählen, um diese Liste zu aktualisieren.

    Wenn Ihre neue Datenquelle in dieser Liste angezeigt wird, kann sie verwendet werden. Sie können „Abfrage“ aus CloudWatch Metriken auswählen, um mit der Abfrage zu beginnen. Weitere Informationen finden Sie unter Erstellen eines Diagramms mit Metriken aus einer anderen Datenquelle.

Amazon Managed Service für Prometheus

Aktualisieren der Datenquellen-Konfiguration

- Sie können mithilfe der folgenden Verfahren Ihre Datenquelle manuell aktualisieren:
  - Um die Workspace-ID von Amazon Managed Service für Prometheus zu aktualisieren, aktualisieren Sie die `AMAZON_PROMETHEUS_WORKSPACE_ID`-Umgebungsvariable für die Lambda-Funktion des Datenquellen-Konnektors.

- Weitere Informationen zum Aktualisieren der VPC-Konfiguration finden Sie unter Konfigurieren des VPC-Zugriffs (Konsole).

Abfragen der Datenquelle

- Bei der Abfrage von Amazon Managed Service für Prometheus können Sie, nachdem Sie die Datenquelle auf der Registerkarte Abfrage mit mehreren Quellen ausgewählt und einen Konnektor von Amazon Managed Service für Prometheus ausgewählt haben, den Abfrageassistenten verwenden, um Metriken und Labels zu ermitteln und einfache PromQL-Abfragen bereitzustellen. Sie können auch den PromQL-Abfragen-Editor verwenden, um eine PromQL-Abfrage zu erstellen.
- Mehrzeilige Abfragen werden von den CloudWatch Datenquellen-Connectoren nicht unterstützt. Jeder Zeilenvorschub wird durch ein Leerzeichen ersetzt, wenn die Abfrage ausgeführt wird oder wenn Sie mit der Abfrage einen Alarm oder ein Dashboard-Widget erstellen. In einigen Fällen kann dies dazu führen, dass Ihre Abfrage ungültig ist. Wenn Ihre Abfrage beispielsweise einen einzeiligen Kommentar enthält, ist sie nicht gültig. Wenn Sie versuchen, ein Dashboard oder einen Alarm mit einer mehrzeiligen Abfrage über die Befehlszeile oder Infrastructure as Code zu erstellen, lehnt die API die Aktion mit einem Analysefehler ab.

OpenSearch Amazon-Dienst

Erstellen der Datenquelle

Wenn die OpenSearch Domain für FGAC aktiviert ist, müssen Sie die Ausführungsrolle der Connector-Lambda-Funktion einem Benutzer in Service zuordnen. OpenSearch Weitere Informationen finden Sie im Abschnitt Benutzer zu Rollen zuordnen unter Verwaltung von Berechtigungen in der OpenSearch Servicedokumentation.

Wenn auf Ihre OpenSearch Domain nur innerhalb einer Virtual Private Cloud (VPC) zugegriffen werden kann, müssen Sie manuell eine neue Umgebungsvariable in die aufgerufene Lambda-Funktion aufnehmen. `AMAZON_OPENSEARCH_ENDPOINT` Der Wert für diese Variable sollte die Stammdomäne des Endpunkts OpenSearch sein. Sie können diese Stammdomäne erhalten, indem Sie `https://` und `<region>.es.amazonaws.com` von dem in der OpenSearch Servicekonsole aufgeführten Domänenendpunkt entfernen. Wenn Ihr Domänenendpunkt beispielsweise lautet`https://sample-domain.us-east-1.es.amazonaws.com`, wäre dies die Stammdomäne`sample-domain`.

Aktualisieren der Datenquelle

- Sie können mithilfe der folgenden Verfahren Ihre Datenquelle manuell aktualisieren:
  - Um die OpenSearch Dienstdomäne zu aktualisieren, aktualisieren Sie die `AMAZON_OPENSEARCH_DOMAIN_NAME` Umgebungsvariable für die Lambda-Funktion des Datenquellenkonnektors.
  - Weitere Informationen zum Aktualisieren der VPC-Konfiguration finden Sie unter Konfigurieren des VPC-Zugriffs (Konsole).

Abfragen der Datenquelle

- Gehen Sie bei der Abfrage von OpenSearch Service nach der Auswahl der Datenquelle auf der Registerkarte Multiquellenabfrage wie folgt vor:
  - Wählen Sie den Index aus, der abgefragt werden soll.
  - Wählen Sie den Metriknamen (beliebiges Zahlenfeld im Dokument) und die Statistik aus.
  - Wählen Sie die Zeitachse aus (beliebiges Datumsfeld im Dokument).
  - Wählen Sie die anzuwendenden Filter aus (Beliebiges Zeichenfolgenfeld im Dokument).
  - Wählen Sie Graph-Abfrage.

Amazon RDS für PostgreSQL und Amazon RDS für MySQL

Erstellen der Datenquelle

- Wenn auf Ihre Datenquelle nur in einer VPC zugegriffen werden kann, müssen Sie die VPC-Konfiguration für den Konnektor angeben, wie unter Mit einem Assistenten eine Verbindung zu einer vordefinierten Datenquelle herstellen beschrieben. Wenn die Datenquelle für Anmeldeinformationen eine Verbindung zur VPC herstellen soll, muss der Endpunkt in der VPC konfiguriert werden. Weitere Informationen finden Sie unter Verwenden eines AWS Secrets Manager VPC-Endpunkts.

  Darüber hinaus müssen Sie einen VPC-Endpunkt für den Amazon RDS-Service erstellen. Weitere Informationen finden Sie unter Amazon RDS-API und VPC-Schnittstellen-Endpunkte ()AWS PrivateLink.

Aktualisieren der Datenquelle

- Sie können mithilfe der folgenden Verfahren Ihre Datenquelle manuell aktualisieren:

- Um die Datenbank-Instance zu aktualisieren, aktualisieren Sie die `RDS_INSTANCE`-Umgebungsvariable für die Lambda-Funktion des Datenquellen-Konnektors.
- Um den Benutzernamen und das Passwort für die Verbindung mit Amazon RDS zu aktualisieren, verwenden Sie AWS Secrets Manager. Sie finden den ARN des für die Datenquelle verwendeten Geheimnisses in der Umgebungsvariable `RDS_SECRET` der Lambda-Funktion der Datenquelle. Weitere Informationen zum Aktualisieren des Geheimnisses in AWS Secrets Manager finden Sie unter Ein AWS Secrets Manager -Geheimnis ändern.
- Weitere Informationen zum Aktualisieren der VPC-Konfiguration finden Sie unter Konfigurieren des VPC-Zugriffs (Konsole).

Abfragen der Datenquelle

- Bei der Abfrage von Amazon RDS können Sie, nachdem Sie die Datenquelle auf der Registerkarte Abfrage mit mehreren Quellen und einen Amazon-RDS-Konnektor ausgewählt haben, den Datenbank-Entdecker verwenden, um verfügbare Datenbanken, Tabellen und Spalten anzuzeigen. Sie können auch den SQL-Editor verwenden, um eine SQL-Abfrage zu erstellen.

  Sie können die folgenden Variablen in der Abfrage verwenden:

  - `$start.iso` – Die Startzeit im ISO-Datumsformat
  - `$end.iso` – Die Endzeit im ISO-Datumsformat
  - `$period` – Der gewählte Zeitraum in Sekunden

  Sie können beispielsweise `SELECT value, timestamp FROM table WHERE timestamp BETWEEN $start.iso and $end.iso` abfragen
- Mehrzeilige Abfragen werden von den CloudWatch Datenquellen-Connectors nicht unterstützt. Jeder Zeilenvorschub wird durch ein Leerzeichen ersetzt, wenn die Abfrage ausgeführt wird oder wenn Sie mit der Abfrage einen Alarm oder ein Dashboard-Widget erstellen. In einigen Fällen kann dies dazu führen, dass Ihre Abfrage ungültig ist. Wenn Ihre Abfrage beispielsweise einen einzeiligen Kommentar enthält, ist sie nicht gültig. Wenn Sie versuchen, ein Dashboard oder einen Alarm mit einer mehrzeiligen Abfrage über die Befehlszeile oder Infrastructure as Code zu erstellen, lehnt die API die Aktion mit einem Analysefehler ab.

> **Note**
>
> Wenn in den Ergebnissen kein Datumsfeld gefunden wird, werden die Werte für jedes Zahlenfeld zu Einzelwerten summiert und über den angegebenen Zeitraum dargestellt. Wenn die Zeitstempel nicht mit dem ausgewählten Zeitraum in übereinstimmen CloudWatch, werden die Daten automatisch anhand des Zeitraums von aggregiert `SUM` und an diesem ausgerichtet. CloudWatch

Amazon-S3-CSV-Dateien

Abfragen der Datenquelle

- Bei der Abfrage von Amazon-S3-CSV-Dateien wählen Sie, nachdem Sie die Datenquelle auf der Registerkarte Abfrage mit mehreren Quellen und einen Amazon-S3-Konnektor ausgewählt haben, den Amazon-S3-Bucket und den Amazon-S3-Schlüssel aus.

  Die CSV-Datei muss wie folgt formatiert werden:

  - Der Zeitstempel muss die erste Spalte sein.
  - Die Tabelle muss eine Kopfzeile haben. Die Überschriften werden verwendet, um Ihre Metriken zu benennen. Der Titel der Zeitstempelspalte wird ignoriert, es werden nur die Titel der Metrikspalten verwendet.
  - Die Zeitstempel müssen im ISO-Datumsformat vorliegen.
  - Bei den Metriken muss es sich um numerische Felder handeln.

```
Timestamp, Metric-1, Metric-2, ...
```

  Im Folgenden wird ein Beispiel gezeigt:

Zeitstempel	CPU (%)	Arbeitsspeicher (%)	Speicher (%)
2023-11-23T17:09:41+00:00	1	2	3
2023-11-23T17:04:41+00:00	4	5	6

Zeitstempel	CPU (%)	Arbeitsspeicher (%)	Speicher (%)
2023-11-23T16:59:41+00:00	7	8	9
2023-11-23T16:54:41+00:00	10	11	12

> **Note**
>
> Wenn kein Zeitstempel angegeben wird, werden die Werte für jede Metrik zu Einzelwerten summiert und über den angegebenen Zeitraum dargestellt. Wenn die Zeitstempel nicht mit dem ausgewählten Zeitraum in übereinstimmen CloudWatch, werden die Daten automatisch anhand des Zeitraums von aggregiert `SUM` und an diesem ausgerichtet. CloudWatch

Microsoft Azure Monitor

Erstellen der Datenquelle

- Sie müssen Ihre Mandanten-ID, Client-ID und Ihr Client-Geheimnis angeben, um eine Verbindung mit Microsoft Azure Monitor herzustellen. Die Anmeldeinformationen werden in gespeichert. AWS Secrets Manager Weitere Informationen finden Sie unter Eine Microsoft-Entra-Anwendung und ein Serviceprinzipal erstellen, die auf Ressourcen zugreifen können in der Microsoft-Dokumentation.

Aktualisieren der Datenquelle

- Sie können mithilfe der folgenden Verfahren Ihre Datenquelle manuell aktualisieren:
  - Um die Mandanten-ID, die Client-ID und den geheimen Client-Schlüssel zu aktualisieren, die für die Verbindung mit Azure Monitor verwendet werden, finden Sie den ARN des für die Datenquelle verwendeten Geheimnisses als `AZURE_CLIENT_SECRET`-Umgebungsvariable in der Lambda-Funktion der Datenquelle. Weitere Informationen zum Aktualisieren des Geheimnisses in AWS Secrets Manager finden Sie unter Ändern eines AWS Secrets Manager Geheimnisses.

Abfragen der Datenquelle

- Bei der Abfrage von Azure Monitor geben Sie, nachdem Sie die Datenquelle auf der Registerkarte Abfrage mit mehreren Quellen und einen Azure-Monitor-Konnektor ausgewählt haben, das Azure-Abonnement sowie die Ressourcengruppe und die Ressource an. Anschließend können Sie den Metrik-Namespace, die Metrik und die Aggregation auswählen und nach Dimensionen filtern.

Prometheus

Erstellen der Datenquelle

- Sie müssen den Prometheus-Endpunkt sowie den Benutzer und das Passwort angeben, die für die Abfrage von Prometheus erforderlich sind. Die Anmeldeinformationen werden in gespeichert AWS Secrets Manager.
- Wenn auf Ihre Datenquelle nur in einer VPC zugegriffen werden kann, müssen Sie die VPC-Konfiguration für den Konnektor angeben, wie unter Mit einem Assistenten eine Verbindung zu einer vordefinierten Datenquelle herstellen beschrieben. Wenn die Datenquelle für Anmeldeinformationen eine Verbindung zur VPC herstellen soll, muss der Endpunkt in der VPC konfiguriert werden. Weitere Informationen finden Sie unter Verwenden eines AWS Secrets Manager VPC-Endpunkts.

Aktualisieren der Datenquellen-Konfiguration

- Sie können mithilfe der folgenden Verfahren Ihre Datenquelle manuell aktualisieren:
  - Um den Prometheus-Endpunkt zu aktualisieren, geben Sie den neuen Endpunkt als `PROMETHEUS_API_ENDPOINT`-Umgebungsvariable in der Lambda-Funktion der Datenquelle an.
  - Um den Benutzernamen und das Passwort für die Verbindung mit Prometheus zu aktualisieren, finden Sie den ARN des für die Datenquelle verwendeten Geheimnisses als `PROMETHEUS_API_SECRET`-Umgebungsvariable in der Lambda-Funktion der Datenquelle. Weitere Informationen zum Aktualisieren des Geheimnisses in finden Sie AWS Secrets Manager unter Ändern eines AWS Secrets Manager Geheimnisses.
  - Weitere Informationen zum Aktualisieren der VPC-Konfiguration finden Sie unter Konfigurieren des VPC-Zugriffs (Konsole).

Abfragen der Datenquelle

> **Important**
>
> Prometheus-Metriktypen unterscheiden sich von CloudWatch Metriken, und viele über Prometheus verfügbare Metriken sind konstruktionsbedingt kumulativ. Wenn Sie Prometheus-Metriken abfragen, wendet CloudWatch keine zusätzliche Transformation auf die Daten an: Wenn Sie nur den Namen oder die Bezeichnung der Metrik angeben, ist der angezeigte Wert kumulativ. Weitere Informationen finden Sie unter Metriktypen in der Prometheus-Dokumentation.
> Um Prometheus-Metrikdaten wie CloudWatch Metriken als diskrete Werte anzuzeigen, müssen Sie die Abfrage bearbeiten, bevor Sie sie ausführen. Sie könnten beispielsweise einen Aufruf der Rate-Funktion über Ihren Prometheus-Metriknamen hinzufügen. Eine Dokumentation zur Rate-Funktion und anderen Prometheus-Funktionen finden Sie unter rate() in der Prometheus-Dokumentation.

Mehrzeilige Abfragen werden von den CloudWatch Datenquellen-Konnektoren nicht unterstützt. Jeder Zeilenvorschub wird durch ein Leerzeichen ersetzt, wenn die Abfrage ausgeführt wird oder wenn Sie mit der Abfrage einen Alarm oder ein Dashboard-Widget erstellen. In einigen Fällen kann dies dazu führen, dass Ihre Abfrage ungültig ist. Wenn Ihre Abfrage beispielsweise einen einzeiligen Kommentar enthält, ist sie nicht gültig. Wenn Sie versuchen, ein Dashboard oder einen Alarm mit einer mehrzeiligen Abfrage über die Befehlszeile oder Infrastructure as Code zu erstellen, lehnt die API die Aktion mit einem Analysefehler ab.

Benachrichtigung über verfügbare Aktualisierungen

Von Zeit zu Zeit informiert Amazon Sie möglicherweise darüber, dass wir empfehlen, Ihre Konnektoren mit einer neueren verfügbaren Version zu aktualisieren und stellt Ihnen Anweisungen dazu zur Verfügung.

Einen Konnektor zu einer Datenquelle erstellen

Um eine Verbindung mit einer benutzerdefinierten Datenquelle herzustellen CloudWatch, haben Sie zwei Möglichkeiten:

- Verwenden Sie zunächst eine Beispielvorlage, die Folgendes CloudWatch bietet: Sie können entweder Python JavaScript oder Python mit dieser Vorlage verwenden. Diese Vorlagen enthalten Lambda-Beispielcode, der Ihnen bei der Erstellung Ihrer Lambda-Funktion nützlich sein wird.

Anschließend können Sie die Lambda-Funktion aus der Vorlage ändern, um eine Verbindung zu Ihrer benutzerdefinierten Datenquelle herzustellen.

- Erstellen Sie eine völlig neue AWS Lambda Funktion, die den Datenquellenkonnektor, die Datenabfrage und die Vorbereitung der Zeitreihen für die Verwendung durch implementiert CloudWatch. Diese Funktion muss Datenpunkte bei Bedarf vorab aggregieren oder zusammenführen und auch den Zeitraum und die Zeitstempel so anpassen, dass sie kompatibel sind. CloudWatch

Inhalt

Eine Vorlage verwenden

Durch die Verwendung einer Vorlage wird eine Lambda-Beispielfunktion erstellt, mit der Sie Ihren benutzerdefinierten Konnektor schneller erstellen können. Diese Beispielfunktionen enthalten Beispielcode für viele gängige Szenarien beim Erstellen eines benutzerdefinierten Konnektors. Sie können den Lambda-Code untersuchen, nachdem Sie einen Konnektor mit einer Vorlage erstellt haben, und ihn dann so ändern, dass er für die Verbindung mit Ihrer Datenquelle verwendet wird.

Wenn Sie die Vorlage verwenden, CloudWatch kümmert er sich außerdem um die Erstellung der Lambda-Berechtigungsrichtlinie und das Anhängen von Ressourcen-Tags an die Lambda-Funktion.

So verwenden Sie die Vorlage, um einen Konnektor für eine benutzerdefinierte Datenquelle zu erstellen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Settings (Einstellungen).
3. Wählen Sie die Registerkarte Metrik-Datenquellen.
4. Klicken Sie auf Create data source.
5. Wählen Sie das Optionsfeld für Benutzerdefiniert – Vorlage für die ersten Schritte aus und dann Weiter.
6. Geben Sie einen Namen für die Datenquelle ein.
7. Wählen Sie eine der aufgelisteten Vorlagen aus.
8. Wählen Sie entweder Node.js oder Python aus.
9. Klicken Sie auf Create data source.

   Die neue benutzerdefinierte Quelle, die Sie gerade hinzugefügt haben, wird erst angezeigt, wenn der AWS CloudFormation Stack ihre Erstellung abgeschlossen hat. Um den Fortschritt zu überprüfen, können Sie „Status meines CloudFormation Stacks anzeigen“ wählen. Oder Sie können das Aktualisierungssymbol wählen, um diese Liste zu aktualisieren.

   Wenn Ihre neue Datenquelle in dieser Liste angezeigt wird, können Sie sie in der Konsole testen und ändern.
10. (Optional) Um die Testdaten aus dieser Quelle in der Konsole abzufragen, folgen Sie den Anweisungen in Erstellen eines Diagramms mit Metriken aus einer anderen Datenquelle.
11. Passen Sie die Lambda-Funktion an Ihre Bedürfnisse an.

    a. Wählen Sie im Navigationsbereich Settings (Einstellungen).

    b. Wählen Sie die Registerkarte Metrik-Datenquellen.

    c. Wählen Sie In Lambda-Konsole anzeigen für die Quelle aus, die Sie ändern möchten.

    Sie können die Funktion jetzt ändern, um auf Ihre Datenquelle zuzugreifen. Weitere Informationen finden Sie unter Schritt 1: Die Funktion erstellen.

> **Note**
>
> Wenn Sie die Vorlage verwenden, müssen Sie beim Schreiben Ihrer Lambda-Funktion nicht den Anweisungen in Schritt 2: Eine Lambda-Berechtigungsrichtlinie erstellen oder Schritt 3: Ein Ressourcen-Tag an die Lambda-Funktion anfügen folgen. Diese Schritte wurden von ausgeführt CloudWatch , weil Sie die Vorlage verwendet haben.

Eine benutzerdefinierte Datenquelle von Grund auf erstellen

Folgen Sie den Schritten in diesem Abschnitt, um eine Lambda-Funktion zu erstellen, die eine Verbindung CloudWatch zu einer Datenquelle herstellt.

Schritt 1: Die Funktion erstellen

Ein benutzerdefinierter Datenquellen-Connector muss `GetMetricData` Ereignisse von CloudWatch unterstützen. Optional können Sie auch ein `DescribeGetMetricData` Ereignis implementieren, um Benutzern in der CloudWatch Konsole eine Dokumentation zur Verwendung des Connectors zur Verfügung zu stellen. Die `DescribeGetMetricData` Antwort kann auch verwendet werden, um Standardwerte festzulegen, die im Generator für CloudWatch benutzerdefinierte Abfragen verwendet werden.

CloudWatch stellt Codefragmente als Beispiele bereit, um Ihnen den Einstieg zu erleichtern. Weitere Informationen finden Sie im Beispiel-Repository unter https://github.com/aws-samples/. cloudwatch-data-source-samples

Beschränkungen

- Die Antwort von Lambda muss kleiner als 6 MB sein. Wenn die Antwort 6 MB überschreitet, markiert die `GetMetricData`-Antwort die Lambda-Funktion als `InternalError` und es werden keine Daten zurückgegeben.
- Die Lambda-Funktion muss die Ausführung innerhalb von 10 Sekunden für Visualisierungs- und Dashboardingzwecke oder innerhalb von 4,5 Sekunden für die Verwendung von Alarmen abschließen. Wenn die Ausführungszeit diesen Wert überschreitet, markiert die `GetMetricData`-Antwort die Lambda-Funktion als `InternalError` und es werden keine Daten zurückgegeben.
- Die Lambda-Funktion muss ihre Ausgabe mit Epochen-Zeitstempeln in Sekunden senden.

- Wenn die Lambda-Funktion die Daten nicht neu berechnet und stattdessen Daten zurückgibt, die nicht der vom CloudWatch Benutzer angeforderten Startzeit und Periodenlänge entsprechen, werden diese Daten von ignoriert. CloudWatch Die zusätzlichen Daten werden bei allen Visualisierungen oder Alarmen verworfen. Alle Daten, die nicht zwischen der Startzeit und der Endzeit liegen, werden ebenfalls verworfen.

  Wenn ein Benutzer beispielsweise nach Daten von 10:00 bis 11:00 Uhr mit einem Zeitraum von 5 Minuten fragt, dann sind „10:00:00 bis 10:04:59“ und „10:05:00 bis 10:09:59“ die gültigen Zeitbereiche für die Rückgabe von Daten. Sie müssen eine Zeitreihe zurückgeben, die `10:00 value1`, `10:05 value2` usw. enthält. Wenn die Funktion beispielsweise `10:03 valueX` zurückgibt, wird sie verworfen, weil 10:03 nicht der angeforderten Startzeit und dem angeforderten Startzeitraum entspricht.
- Mehrzeilige Abfragen werden von den CloudWatch Datenquellen-Connectors nicht unterstützt. Jeder Zeilenvorschub wird durch ein Leerzeichen ersetzt, wenn die Abfrage ausgeführt wird oder wenn Sie mit der Abfrage einen Alarm oder ein Dashboard-Widget erstellen. In einigen Fällen kann dies dazu führen, dass Ihre Abfrage ungültig ist.

GetMetricData Ereignis

Anforderungs-Nutzlast

Im Folgenden sehen Sie ein Beispiel für eine `GetMetricData`-Anforderungs-Nutzlast, die als Eingabe an die Lambda-Funktion gesendet wird.

```
{
  "EventType": "GetMetricData",
  "GetMetricDataRequest": {
    "StartTime": 1697060700,
    "EndTime": 1697061600,
    "Period": 300,
    "Arguments": ["serviceregistry_external_http_requests{host_cluster!=\"prod\"}"]
  }
}
```

- StartTime— Der Zeitstempel, der die frühesten zurückzugebenden Daten angibt. Der Typ ist Zeitstempel, Epoche, Sekunden.
- EndTime— Der Zeitstempel, der die letzten zurückzugebenden Daten angibt. Der Typ ist Zeitstempel, Epoche, Sekunden.

- Zeitraum – Die Anzahl der Sekunden, die jede Aggregation der Metrikdaten darstellt. Der Mindestwert beträgt 60 Sekunden. Der Typ ist Sekunden.
- Argumente – Ein Array von Argumenten, die an den mathematischen Ausdruck der Lambda-Metrik übergeben werden. Informationen zur Übergabe von Argumenten finden Sie unter So übergeben Sie Argumente an Ihre Lambda-Funktion.

Antwort-Nutzlast

Nachfolgend sehen Sie ein Beispiel für eine von der Lambda-Funktion zurückgegebenen `GetMetricData`-Antwort-Nutzlast.

```
{
   "MetricDataResults": [
      {
         "StatusCode": "Complete",
         "Label": "CPUUtilization",
         "Timestamps": [ 1697060700, 1697061000, 1697061300 ],
         "Values": [ 15000, 14000, 16000 ]
      }
   ]
}
```

Die Antwort-Nutzlast enthält entweder ein `MetricDataResults`-Feld oder ein `Error`-Feld, aber nicht beides.

Ein `MetricDataResults`-Feld ist eine Liste von Zeitreihenfeldern des Typs `MetricDataResult`. Jedes dieser Zeitreihenfelder kann die folgenden Felder enthalten.

- StatusCode— (Optional) `Complete` gibt an, dass alle Datenpunkte im angeforderten Zeitraum zurückgegeben wurden. `PartialDatabedeutet`, dass ein unvollständiger Satz von Datenpunkten zurückgegeben wurde. Wenn dieses Argument weggelassen wird, ist der Standardwert `Complete`.

  Zulässige Werte: `Complete` | `InternalError` | `PartialData` | `Forbidden`
- Nachrichten – Optionale Liste von Nachrichten mit zusätzlichen Informationen zu den zurückgegebenen Daten.

  Typ: Anordnung von MessageDataObjekten mit `Code` und `Value` -Zeichenketten.
- Label – Das für Menschen lesbare Etikett, das den Daten zugeordnet ist.

  Typ: Zeichenfolge

- Zeitstempel – Die Zeitstempel für die Datenpunkte, formatiert in Epochenzeit. Die Anzahl der Zeitstempel entspricht immer der Anzahl der Werte und der Wert für `Timestamps[x]` ist `Values[x]`.

  Typ: Array von Zeitstempeln

- Werte – Die Datenpunktwerte für die Metrik, entsprechend der `Timestamps`. Die Anzahl der Werte entspricht immer der Anzahl der Zeitstempel und der Wert für `Timestamps[x]` ist `Values[x]`.

  Typ: Array von Dubletten

Weitere Informationen über `Error`-Objekte finden Sie in den folgenden Abschnitten.

Formate für die Fehlerantwort

Sie können optional die Fehlerantwort verwenden, um weitere Informationen zu Fehlern bereitzustellen. Wir empfehlen, dass Sie mit der Codevalidierung einen Fehler zurückgeben, wenn ein Validierungsfehler auftritt, z. B. wenn ein Parameter fehlt oder vom falschen Typ ist.

Im Folgenden finden Sie ein Beispiel für die Reaktion, wenn die Lambda-Funktion eine `GetMetricData`-Validierungsausnahme auslösen möchte.

```
{
   "Error": {
      "Code": "Validation",
      "Value": "Invalid Prometheus cluster"
   }
}
```

Das Folgende ist ein Beispiel für die Reaktion, wenn die Lambda-Funktion angibt, dass sie aufgrund eines Zugriffsproblems keine Daten zurückgeben kann. Die Antwort wird in eine einzige Zeitreihe mit dem Statuscode `Forbidden` übersetzt.

```
{
   "Error": {
      "Code": "Forbidden",
      "Value": "Unable to access ..."
   }
}
```

Das Folgende ist ein Beispiel dafür, wann die Lambda-Funktion eine allgemeine `InternalError`-Ausnahme auslöst, die in eine einzige Zeitreihe mit dem Statuscode `InternalError` und einer Nachricht übersetzt wird. Wenn ein Fehlercode einen anderen Wert als `Validation` oder hat`Forbidden`, CloudWatch wird davon ausgegangen, dass es sich um einen generischen internen Fehler handelt.

```
{
   "Error": {
      "Code": "PrometheusClusterUnreachable",
      "Value": "Unable to communicate with the cluster"
   }
}
```

DescribeGetMetricData Ereignis

Anforderungs-Nutzlast

Es folgt ein Beispiel für eine `DescribeGetMetricData`-Anforderungs-Nutzlast.

```
{
  "EventType": "DescribeGetMetricData"
}
```

Antwort-Nutzlast

Es folgt ein Beispiel für eine `DescribeGetMetricData`-Antwort-Nutzlast.

```
{
    "Description": "Data source connector",
    "ArgumentDefaults": [{
        Value: "default value"
     }]
}
```

- Beschreibung – Eine Beschreibung der Verwendung des Datenquellen-Konnektors. Diese Beschreibung wird in der CloudWatch Konsole angezeigt. Markdown wird unterstützt.

  Typ: Zeichenfolge
- ArgumentDefaults— Optionales Array von Argumentstandardwerten, die verwendet werden, um den benutzerdefinierten Datenquellen-Builder vorab auszufüllen.

Wenn`[{ Value: "default value 1"}, { Value: 10}]`, zurückgegeben wird, zeigt der Query Builder in der CloudWatch Konsole zwei Eingaben an, die erste mit „Standardwert 1“ und die zweite mit 10.

Wenn `ArgumentDefaults` nicht angegeben wird, wird eine einzelne Eingabe angezeigt, wobei der Standardtyp auf `String` gesetzt ist.

Typ: Array von Objekten, die Wert und Typ enthalten.

- Fehler – (Optional) In jeder Antwort kann ein Fehlerfeld enthalten sein. Beispiele finden Sie unter GetMetricData Ereignis.

Wichtige Überlegungen zu CloudWatch Alarmen

Wenn Sie die Datenquelle zum Einstellen von CloudWatch Alarmen verwenden möchten, sollten Sie sie so einrichten, dass Daten mit Zeitstempeln pro Minute bis CloudWatch gemeldet werden. Weitere Informationen und weitere Überlegungen zur Erstellung von Alarmen für Metriken aus verbundenen Datenquellen finden Sie unter Einen Alarm basierend auf einer verbundenen Datenquelle erstellen.

(Optional) Wird AWS Secrets Manager zum Speichern von Anmeldeinformationen verwendet

Wenn Ihre Lambda-Funktion Anmeldeinformationen für den Zugriff auf die Datenquelle verwenden muss, empfehlen wir, diese Anmeldeinformationen AWS Secrets Manager zu speichern, anstatt sie fest in Ihre Lambda-Funktion zu codieren. Weitere Informationen zur Verwendung AWS Secrets Manager mit Lambda finden Sie unter Verwenden von AWS Secrets Manager Geheimnissen in AWS Lambda Funktionen.

(Optional) Mit einer Datenquelle in einer VPC verbinden

Wenn sich Ihre Datenquelle in einer von Amazon Virtual Private Cloud verwalteten VPC befindet, müssen Sie Ihre Lambda-Funktion für den Zugriff darauf konfigurieren. Weitere Informationen finden Sie unter Verbinden von ausgehenden Netzwerken mit Ressourcen in einer VPC.

Möglicherweise müssen Sie auch VPC-Service-Endpunkte für den Zugriff auf Services wie AWS Secrets Manager konfigurieren. Weitere Informationen finden Sie unter Zugreifen auf einen AWS Dienst über einen Schnittstellen-VPC-Endpunkt.

Schritt 2: Eine Lambda-Berechtigungsrichtlinie erstellen

Sie müssen eine Richtlinienanweisung erstellen, die die CloudWatch Erlaubnis erteilt, die von Ihnen erstellte Lambda-Funktion zu verwenden. Sie können die AWS CLI oder die Lambda-Konsole verwenden, um die Richtlinienerklärung zu erstellen.

Um die Richtlinienerklärung AWS CLI zu erstellen

- Geben Sie den folgenden Befehl ein. Ersetzen Sie *123456789012* durch Ihre Konto-ID, *my-data-source-function*ersetzen Sie durch den Namen Ihrer Lambda-Funktion und ersetzen Sie *MyDataSource- DataSourcePermission 1234* durch einen beliebigen eindeutigen Wert.

```
aws lambda add-permission --function-name my-data-source-function --statement-
id MyDataSource-DataSourcePermission1234 --action lambda:InvokeFunction --principal
 lambda.datasource.cloudwatch.amazonaws.com --source-account 123456789012
```

Schritt 3: Ein Ressourcen-Tag an die Lambda-Funktion anfügen

Die CloudWatch Konsole bestimmt mithilfe eines Tags, welche Ihrer Lambda-Funktionen Datenquellen-Konnektoren sind. Wenn Sie mit einem der Assistenten eine Datenquelle erstellen, wird das Tag automatisch von dem AWS CloudFormation Stack angewendet, der es konfiguriert. Wenn Sie selbst eine Datenquelle erstellen, können Sie das folgende Tag für Ihre Lambda-Funktion verwenden. Dadurch wird Ihr Connector in der Dropdownliste Datenquelle in der CloudWatch Konsole angezeigt, wenn Sie Metriken abfragen.

- Ein Tag mit `cloudwatch:datasource` als Schlüssel und `custom` als Wert.

Ihre benutzerdefinierte Datenquelle verwenden

Nachdem Sie eine Datenquelle erstellt haben, können Sie sie verwenden, um Daten aus dieser Quelle abzufragen, sie zu visualisieren und Alarme einzustellen. Wenn Sie die Vorlage verwendet haben, um Ihren benutzerdefinierten Datenquellen-Konnektor zu erstellen, oder wenn Sie das unter Schritt 3: Ein Ressourcen-Tag an die Lambda-Funktion anfügen aufgeführte Tag hinzugefügt haben, können Sie die unter Erstellen eines Diagramms mit Metriken aus einer anderen Datenquelle aufgeführten Schritte zum Abfragen ausführen.

Sie können den Konnektor auch mit der mathematischen Metrikfunktion LAMBDA abfragen, wie im folgenden Abschnitt beschrieben.

Informationen zum Erstellen von Alarmen für eine Metrik aus Ihrer Datenquelle finden Sie unter Einen Alarm basierend auf einer verbundenen Datenquelle erstellen.

So übergeben Sie Argumente an Ihre Lambda-Funktion

Es wird empfohlen, Argumente an Ihre benutzerdefinierte Datenquelle zu übergeben, indem Sie den Query Builder in der CloudWatch Konsole verwenden, wenn Sie die Datenquelle abfragen.

Sie können Ihre Lambda-Funktion auch verwenden, um Daten aus Ihrer Datenquelle abzurufen, indem Sie den neuen LAMBDA Ausdruck in CloudWatch metrischer Mathematik verwenden.

```
LAMBDA("LambdaFunctionName" [, optional-arg]*)
```

`optional-arg` besteht aus bis zu 20 Zeichenfolgen, Zahlen oder Booleschen Werten. Beispiel: `param`, `3.14` oder `true`.

> **Note**
>
> Mehrzeilige Zeichenfolgen werden von den CloudWatch Datenquellenkonnektoren nicht unterstützt. Jeder Zeilenvorschub wird durch ein Leerzeichen ersetzt, wenn die Abfrage ausgeführt wird oder wenn Sie mit der Abfrage einen Alarm oder ein Dashboard-Widget erstellen. In einigen Fällen kann dies dazu führen, dass Ihre Abfrage ungültig ist.

Wenn Sie die mathematische LAMBDA-Metrikfunktion verwenden, können Sie den Funktionsnamen (`"MyFunction"`) angeben. Wenn Ihre Ressourcenrichtlinie dies zulässt, können Sie auch eine bestimmte Version der Funktion (`"MyFunction:22"`) oder einen Lambda-Funktionsalias (`"MyFunction:MyAlias"`) verwenden. Sie können keinen * verwenden

Im Folgenden werden einige Beispiele für das Aufrufen der LAMBDA-Funktion aufgeführt.

```
LAMBDA("AmazonOpenSearchDataSource", "MyDomain", "some-query")
```

```
LAMBDA("MyCustomDataSource", true, "fuzzy", 99.9)
```

Die mathematische LAMBDA-Metrikfunktion gibt eine Liste von Zeitreihen zurück, die an den Anforderer zurückgegeben oder mit anderen mathematischen Metrikfunktionen kombiniert werden können. Im Folgenden finden Sie ein Beispiel für die Kombination von LAMBDA mit anderen mathematischen Metrikfunktionen.

```
FILL(LAMBDA("AmazonOpenSearchDataSource", "MyDomain", "some-query"), 0)
```

Den Konnektor einer Datenquelle löschen

Um den Konnektor einer Datenquelle zu löschen, folgen Sie den Anweisungen in diesem Abschnitt.

So löschen Sie den Konnektor einer Datenquelle

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Settings (Einstellungen).
3. Wählen Sie die Registerkarte Metrik-Datenquellen.
4. Wählen Sie CloudFormationin der Zeile der Datenquelle, die Sie löschen möchten, die Option Verwalten aus.

   Sie werden zur AWS CloudFormation Konsole weitergeleitet.
5. Wählen Sie im Abschnitt mit dem Namen Ihrer Datenquelle die Option Löschen.
6. Wählen Sie im Bestätigungsfenster Löschen.

Erfassen Sie mit dem CloudWatch Agenten Metriken, Logs und Traces

Mit dem vereinheitlichten CloudWatch Agenten können Sie Folgendes tun:

- Erfassen Sie intern betriebssystemübergreifend mehr Metriken auf Systemebene von Amazon-EC2-Instances. Die Metriken können neben den Metriken für EC2-Instances auch Gast-Metriken enthalten. Eine Aufstellung der zusätzlichen Metriken, die erfasst werden können, finden Sie unter Vom CloudWatch Agenten gesammelte Metriken.
- Erfassen Sie Metriken auf Systemebene von On-Premises-Servern. Dazu können sowohl Server in einer Hybridumgebung als auch Server gehören, die nicht von verwaltet werden AWS.
- Rufen Sie benutzerdefinierte Metriken aus Ihren Anwendungen oder Services mithilfe der Protokolle `StatsD` und `collectd` ab. `StatsD` wird sowohl auf Linux-Servern als auch auf Servern mit Windows Server unterstützt. `collectd` wird nur auf Linux-Servern unterstützt.
- Erfassen Sie Protokolle von Amazon-EC2-Instances und On-Premises-Servern mit Linux oder Windows Server.

  > **Note**
  >
  > Der CloudWatch Agent unterstützt das Sammeln von Protokollen aus FIFO-Pipes nicht.

- Version 1.300031.0 und höher können verwendet werden, um Application Signals zu aktivieren. CloudWatch Weitere Informationen finden Sie unter Application Signals.
- Version 1.300025.0 und höher können Traces von OpenTelemetryoder X-Ray-Client-SDKs sammeln und an X-Ray senden.

  Mit dem CloudWatch Agenten können Sie Traces sammeln, ohne einen separaten Trace-Collection-Daemon ausführen zu müssen, wodurch die Anzahl der Agents, die Sie ausführen und verwalten, reduziert wird.

Sie können die Messwerte, die Sie mit dem CloudWatch Agenten sammeln, CloudWatch genauso speichern und anzeigen wie alle anderen CloudWatch Messwerte. Der Standard-Namespace für die vom CloudWatch Agenten gesammelten Metriken lautet`CWAgent`, obwohl Sie bei der Konfiguration des Agenten einen anderen Namespace angeben können.

Die vom Unified CloudWatch Agent gesammelten Protokolle werden verarbeitet und in Amazon CloudWatch Logs gespeichert, genau wie die vom älteren Logs-Agenten gesammelten CloudWatch Protokolle. Informationen zu den Preisen für CloudWatch Logs finden Sie unter CloudWatch Amazon-Preise.

Die vom CloudWatch Agenten gesammelten Metriken werden als benutzerdefinierte Metriken in Rechnung gestellt. Weitere Informationen zu den Preisen von CloudWatch Metriken finden Sie unter CloudWatchAmazon-Preise.

Der CloudWatch Agent ist unter der MIT-Lizenz als Open-Source-Software verfügbar und wird auf GitHub gehostet. Wenn Sie den CloudWatch Agenten erstellen, anpassen oder dazu beitragen möchten, finden Sie die neuesten Anweisungen im GitHub Repository. Wenn Sie glauben, ein potenzielles Sicherheitsproblem entdeckt zu haben, posten Sie es nicht in einem öffentlichen Forum. GitHub Folgen Sie stattdessen den Anweisungen unter Schwachstellenmeldung oder wenden Sie sich direkt an die AWS E-Mail-Sicherheit.

In den Schritten in diesem Abschnitt wird erklärt, wie der Unified CloudWatch Agent auf Amazon EC2 EC2-Instances und lokalen Servern installiert wird. Weitere Informationen zu den Metriken, die der CloudWatch Agent sammeln kann, finden Sie unter. Vom CloudWatch Agenten gesammelte Metriken

Unterstützte Betriebssysteme

Der CloudWatch Agent wird auf der x86-64-Architektur unter den folgenden Betriebssystemen unterstützt. Er wird auch bei allen kleineren Versionsupdates für jede der hier aufgeführten Hauptversionen unterstützt.

- Amazon Linux 2023
- Amazon Linux 2
- Ubuntu Server-Versionen 23.10, 22.04, 20.04, 18.04, 16.04 und 14.04
- CentOS-Versionen 9, 8 und 7
- Red Hat Enterprise Linux (RHEL)-Versionen 9, 8 und 7
- Debian-Versionen 12, 11 und 10
- SUSE Linux Enterprise Server (SLES)-Versionen 15 und 12
- Oracle Linux-Versionen 9, 8 und 7
- AlmaLinux Versionen 9 und 8
- Rocky-Linux-Versionen 9 und 8

- Die folgenden macOS-Computer: EC2 M1 Mac1-Instances und Computer mit macOS 14 (Sonoma), macOS 13 (Ventura) und macOS 12 (Monterey)
- 64-Bit-Versionen von Windows Server 2022, Windows Server 2019 und Windows Server 2016
- Windows 10 (64 Bit)

Der Agent wird von ARM64-Architektur auf folgenden Betriebssystemen unterstützt. Er wird auch bei allen kleineren Versionsupdates für jede der hier aufgeführten Hauptversionen unterstützt.

- Amazon Linux 2023
- Amazon Linux 2
- Ubuntu Server-Versionen 23.10, 22.04, 20.04, 18.04 und 16.04
- CentOS-Versionen 9 und 8
- Red Hat Enterprise Linux (RHEL)-Versionen 9, 8 und 7
- Debian-Versionen 12, 11 und 10
- SUSE Linux Enterprise Server 15
- Die folgenden macOS-Computer: macOS 14 (Sonoma), macOS 13 (Ventura) und macOS 12 (Monterey)

Übersicht über den Installationsprozess

Sie können den CloudWatch Agenten manuell über die Befehlszeile herunterladen und installieren oder ihn in SSM integrieren. Der allgemeine Ablauf der Installation des CloudWatch Agenten mit einer der beiden Methoden sieht wie folgt aus:

1. Erstellen Sie IAM-Rollen oder -Benutzer, die es dem Agenten ermöglichen, Metriken vom Server zu sammeln und optional in diese zu integrieren AWS Systems Manager.
2. Laden Sie das Agentenpaket herunter.
3. Ändern Sie die CloudWatch Agent-Konfigurationsdatei und geben Sie die Metriken an, die Sie sammeln möchten.
4. Installieren und starten Sie den Agenten auf Ihren Servern. Fügen Sie beim Installieren des Agenten auf einer EC2-Instance die IAM-Rolle hinzu, die Sie in Schritt 1 erstellt haben. Geben Sie beim Installieren des Agenten auf einem On-Premises-Server ein benanntes Profil mit den Anmeldeinformationen des IAM-Benutzers an, den Sie in Schritt 1 erstellt haben.

Inhalt

Den CloudWatch Agenten installieren

Der CloudWatch Agent ist als Paket in Amazon Linux 2023 und Amazon Linux 2 verfügbar. Wenn Sie eines dieser Betriebssysteme verwenden, können Sie das Paket installieren, indem Sie den folgenden Befehl eingeben. Sie müssen außerdem sicherstellen, dass der IAM-Rolle, die der Instanz zugewiesen ist, die CloudWatchAgentServerPolicyangehängt ist. Weitere Informationen finden Sie unter Erstellen Sie IAM-Rollen zur Verwendung mit dem CloudWatch Agenten auf Amazon EC2 EC2-Instances.

```
sudo yum install amazon-cloudwatch-agent
```

Auf allen unterstützten Betriebssystemen, einschließlich Linux und Windows Server, können Sie den CloudWatch Agenten entweder über die Befehlszeile mit einem Amazon S3 S3-Download-Link, mithilfe von Amazon EC2 Systems Manager oder mithilfe einer AWS CloudFormation Vorlage herunterladen und installieren. Ausführliche Informationen finden Sie in den folgenden Abschnitten:

Inhalt

Den CloudWatch Agenten über die Befehlszeile installieren

Verwenden Sie die folgenden Themen, um das CloudWatch Agentenpaket herunterzuladen, zu konfigurieren und zu installieren.

Themen

Laden Sie den CloudWatch Agenten über die Befehlszeile herunter und konfigurieren Sie ihn

Gehen Sie wie folgt vor, um das CloudWatch Agentenpaket herunterzuladen, IAM-Rollen oder -Benutzer zu erstellen und optional die allgemeine Konfigurationsdatei zu ändern.

Laden Sie das CloudWatch Agentenpaket herunter

> **Note**
>
> Um den CloudWatch Agenten herunterzuladen, muss Ihre Verbindung TLS 1.2 oder höher verwenden.

Der CloudWatch Agent ist als Paket in Amazon Linux 2023 und Amazon Linux 2 verfügbar. Wenn Sie dieses Betriebssystem verwenden, können Sie das Paket installieren, indem Sie den folgenden Befehl eingeben. Sie müssen außerdem sicherstellen, dass der IAM-Rolle, die der Instance zugewiesen ist, die CloudWatchAgentServerPolicyangehängt ist. Weitere Informationen finden Sie unter Erstellen Sie IAM-Rollen und -Benutzer für die Verwendung mit dem Agenten CloudWatch .

```
sudo yum install amazon-cloudwatch-agent
```

Auf allen unterstützten Betriebssystemen können Sie den CloudWatch Agenten über die Befehlszeile herunterladen und installieren.

Für jeden Download-Link gibt es einen allgemeinen Link sowie Links für jede Region. Für Amazon Linux 2023 und Amazon Linux 2 und die x86-64-Architektur lauten beispielsweise drei der gültigen Download-Links:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-us-east-1.s3.us-east-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-eu-central-1.s3.eu-central-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

Sie können auch eine README-Datei über die neuesten Änderungen am Agenten sowie eine Datei mit der Versionsnummer herunterladen, die zum Download verfügbar ist. Diese Dateien befinden sich an den folgenden Speicherorten:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/RELEASE_NOTES` oder `https://amazoncloudwatch-agent-`*`region`*`.s3.`*`region`*`.amazonaws.com/info/latest/RELEASE_NOTES`
- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/CWAGENT_VERSION` oder `https://amazoncloudwatch-agent-`*`region`*`.s3.`*`region`*`.amazonaws.com/info/latest/CWAGENT_VERSION`

Architektur	Plattform	Download-Link	Link zur Signaturdatei
x86-64	Amazon Linux 2023 und Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/ .rpm amazon-cloudwatch-agent https://amazoncloudwatch-agent — *Region .s3.region .amazonaws.com/amazon_linux/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3.region .amazonaws.com/amazon_linux/amd64/latest/* .rpm.sig amazon-cloudwatch-agent

Architektur	Plattform	Download-Link	Link zur Signaturdatei
x86-64	Centos	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/centos/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/centos/amd64/latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/redhat/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/redhat/amd64/latest/* .rpm.sig amazon-cloudwatch-agent

Architektur	Plattform	Download-Link	Link zur Signaturdatei
x86-64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/suse/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/suse/amd64/latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	Debian	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/debian/amd64/latest/* .deb amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/ amazon-cloudwatch-agent .deb.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/debian/amd64/latest/* .deb.sig amazon-cloudwatch-agent

Architektur	Plattform	Download-Link	Link zur Signaturdatei
x86-64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/ubuntu/amd64/latest/* .deb amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/ amazon-cloudwatch-agent .deb.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/ubuntu/amd64/latest/* .deb.sig amazon-cloudwatch-agent
x86-64	Oracle	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/oracle_linux/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/oracle_linux/amd64/latest/* .rpm.sig amazon-cloudwatch-agent

Architektur	Plattform	Download-Link	Link zur Signaturdatei
x86-64	macOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/ amazon-cloudwatch-agent .pkg https://amazoncloudwatch-agent - *Region .s3. region .amazonaws.com/darwin/amd64 /latest/* .pkg amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/ amazon-cloudwatch-agent .pkg.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/darwin/amd64/latest/* .pkg.sig amazon-cloudwatch-agent
x86-64	Windows	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/ amazon-cloudwatch-agent .msi https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/windows/amd64/latest/* .msi amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/ amazon-cloudwatch-agent .msi.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/windows/amd64/latest/* .msi.sig amazon-cloudwatch-agent

Architektur	Plattform	Download-Link	Link zur Signaturdatei
ARM64	Amazon Linux 2023 und Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3.region .amazonaws.com/amazon_linux/arm64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3.region .amazonaws.com/amazon_linux/arm64/latest/* .rpm.sig amazon-cloudwatch-agent
ARM64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3.region .amazonaws.com/redhat/arm64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3.region .amazonaws.com/redhat/arm64/latest/* .rpm.sig amazon-cloudwatch-agent

Architektur	Plattform	Download-Link	Link zur Signaturdatei
ARM64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/ubuntu/arm64/latest/* .deb amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/ amazon-cloudwatch-agent .deb.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/ubuntu/arm64/latest/* .deb.sig amazon-cloudwatch-agent
ARM64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/suse/arm64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/suse/arm64/latest/* .rpm.sig amazon-cloudwatch-agent

Architektur	Plattform	Download-Link	Link zur Signaturdatei
ARM64	MacOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/arm64/latest/ amazon-cloudwatch-agent .pkg https://amazoncloudwatch-agent - *Region .s3. region .amazonaws.com/darwin/arm64/latest/* .pkg amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/arm64/latest/ amazon-cloudwatch-agent .pkg.sig *https://amazoncloudwatch-agent – Region .s3. region .amazonaws.com/darwin/arm64/latest/* .pkg.sig amazon-cloudwatch-agent

Um das Agentenpaket über die Befehlszeile herunterzuladen und zu installieren CloudWatch

1. Laden Sie den CloudWatch Agenten herunter.

   Geben Sie auf einem Linux-Server Folgendes ein. Bei *download-link* verwenden Sie den entsprechenden Download-Link aus der vorherigen Tabelle.

   ```
   wget download-link
   ```

   Laden Sie für einen Server mit Windows Server die folgende Datei herunter:

   ```
   https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi
   ```

2. Nachdem Sie das Paket heruntergeladen haben, können Sie optional die Paketsignatur überprüfen. Weitere Informationen finden Sie unter Überprüfung der Signatur des Agentenpakets CloudWatch .

3. Installieren Sie das Paket . Wenn Sie ein RPM-Paket auf einen Linux-Server heruntergeladen haben, wechseln Sie in das Verzeichnis, das das Paket enthält, und geben Sie Folgendes ein:

   ```
   sudo rpm -U ./amazon-cloudwatch-agent.rpm
   ```

Wenn Sie ein DEB-Paket auf einen Linux-Server heruntergeladen haben, wechseln Sie in das Verzeichnis, das das Paket enthält, und geben Sie Folgendes ein:

```
sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
```

Wenn Sie ein MSI-Paket auf einem Server mit Windows Server heruntergeladen haben, wechseln Sie in das Verzeichnis, das das Paket enthält, und geben Sie Folgendes ein:

```
msiexec /i amazon-cloudwatch-agent.msi
```

Dieser Befehl funktioniert auch von innen heraus PowerShell. Weitere Informationen zu den MSI-Befehlsoptionen finden Sie unter Command-Line Options in der Microsoft Windows-Dokumentation.

Wenn Sie ein PKG-Paket auf einen macOS-Server heruntergeladen haben, wechseln Sie in das Verzeichnis, das das Paket enthält, und geben Sie Folgendes ein:

```
sudo installer -pkg ./amazon-cloudwatch-agent.pkg -target /
```

Erstellen und Ändern der Agentenkonfigurationsdatei

Nachdem Sie den CloudWatch Agenten heruntergeladen haben, müssen Sie die Konfigurationsdatei erstellen, bevor Sie den Agenten auf beliebigen Servern starten. Weitere Informationen finden Sie unter Erstellen Sie die CloudWatch Agent-Konfigurationsdatei.

Erstellen Sie IAM-Rollen und -Benutzer für die Verwendung mit dem Agenten CloudWatch

Für den Zugriff auf AWS Ressourcen sind Berechtigungen erforderlich. Sie erstellen eine IAM-Rolle, einen IAM-Benutzer oder beides, um Berechtigungen zu erteilen, für die der CloudWatch Agent Metriken schreiben muss. CloudWatch Wenn Sie den Agenten auf Amazon-EC2-Instances verwenden, müssen Sie eine IAM-Rolle erstellen. Wenn Sie den Agenten auf On-Premises-Servern verwenden, müssen Sie einen IAM-Benutzer erstellen.

> **Note**
>
> Wir haben die folgenden Verfahren kürzlich geändert, indem die neuen, von Amazon erstellten Richtlinien `CloudWatchAgentServerPolicy` und `CloudWatchAgentAdminPolicy` verwendet werden, anstatt dass Kunden diese Richtlinien selbst erstellen müssen. Die von Amazon erstellten Richtlinien unterstützen für das Schreiben von Dateien zu und das Herunterladen von Dateien von Parameter Store nur Dateien, deren Namen mit `AmazonCloudWatch-` beginnen. Wenn Sie über eine CloudWatch Agentenkonfigurationsdatei verfügen, deren Dateiname nicht mit beginnt`AmazonCloudWatch-`, können diese Richtlinien nicht verwendet werden, um die Datei in den Parameter Store zu schreiben oder aus dem Parameter Store herunterzuladen.

Wenn Sie den CloudWatch Agenten auf Amazon EC2 EC2-Instances ausführen möchten, gehen Sie wie folgt vor, um die erforderliche IAM-Rolle zu erstellen. Diese Rolle bietet Berechtigungen zum Lesen und Schreiben von Informationen aus der Instance. CloudWatch

Um die IAM-Rolle zu erstellen, die für die Ausführung des CloudWatch Agenten auf EC2-Instances erforderlich ist

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie links im Navigationsbereich Roles (Rollen) und dann Create Role (Rolle erstellen) aus.
3. Stellen Sie sicher, dass unter Trusted entity type (Typ der vertrauenswürdigen Entität auswählen der AWS -Service ausgewählt ist.
4. Wählen Sie für Anwendungsfall die Option EC2 unter Häufige Anwendungsfälle.
5. Wählen Sie Weiter aus.
6. Aktivieren Sie in der Liste der Richtlinien das Kontrollkästchen neben CloudWatchAgentServerPolicy. Verwenden Sie ggf. das Suchfeld, um die Richtlinie zu finden.
7. (Optional) Wenn der Agent Traces an X-Ray sendet, müssen Sie der Rolle auch die AWSXRayDaemonWriteAccessRichtlinie geben. Suchen Sie dazu diese Richtlinie in der Liste und aktivieren Sie das Kontrollkästchen daneben.
8. Wählen Sie Weiter aus.

9. Geben Sie im Feld Rollenname einen Namen für die Rolle ein, z. *CloudWatchAgentServerRole*B. Geben Sie optional eine Beschreibung dafür ein. Wählen Sie dann Create Role.

   Die Rolle wird jetzt erstellt.

10. (Optional) Wenn der Agent Protokolle an Logs senden soll und Sie möchten, dass der Agent Aufbewahrungsrichtlinien für diese Protokollgruppen festlegen kann, müssen Sie der Rolle die `logs:PutRetentionPolicy` entsprechende Berechtigung hinzufügen. CloudWatch Weitere Informationen finden Sie unter Erlauben Sie dem CloudWatch Agenten, eine Richtlinie zur Aufbewahrung von Protokollen festzulegen.

Wenn Sie den CloudWatch Agenten auf lokalen Servern ausführen möchten, gehen Sie wie folgt vor, um den erforderlichen IAM-Benutzer zu erstellen.

> **Warning**
>
> Für dieses Szenario sind IAM-Benutzer mit programmatischem Zugriff und langfristigen Anmeldeinformationen erforderlich, was ein Sicherheitsrisiko darstellt. Um dieses Risiko zu minimieren, empfehlen wir, diesen Benutzern nur die Berechtigungen zu gewähren, die sie für die Ausführung der Aufgabe benötigen, und diese Benutzer zu entfernen, wenn sie nicht mehr benötigt werden. Die Zugriffsschlüssel können bei Bedarf aktualisiert werden. Weitere Informationen finden Sie unter Aktualisieren von Zugriffsschlüsseln im IAM-Benutzerhandbuch.

So erstellen Sie den IAM-Benutzer, der für die Ausführung des CloudWatch Agenten auf lokalen Servern erforderlich ist

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Klicken Sie im Navigationsbereich links auf Benutzer und dann auf Benutzer hinzufügen.
3. Geben Sie den Benutzernamen für den neuen Benutzer an.
4. Wählen Sie Access key - Programmatic access (Zugriffsschlüssel – programmgesteuerter Zugriff) und Next: Permissions (Weiter: Berechtigungen) aus.
5. Wählen Sie Vorhandene Richtlinien direkt zuordnen.

6. Aktivieren Sie in der Liste der Richtlinien das Kontrollkästchen neben CloudWatchAgentServerPolicy. Verwenden Sie ggf. das Suchfeld, um die Richtlinie zu finden.
7. (Optional) Wenn der Agent Traces zu X-Ray weiterleiten will, müssen Sie der Rolle auch die AWSXRayDaemonWriteAccessRichtlinie geben. Suchen Sie dazu diese Richtlinie in der Liste und aktivieren Sie das Kontrollkästchen daneben.
8. Wählen Sie Weiter: Markierungen.
9. Erstellen Sie optional Tags für den neuen IAM-Benutzer und wählen Sie dann Next: Review (Weiter: Prüfung) aus.
10. Prüfen Sie, ob die richtigen Richtlinie aufgelistet ist, und klicken Sie auf Benutzer erstellen.
11. Klicken Sie neben dem Namen des neuen Benutzers auf Show. Kopieren Sie den Zugriffsschlüssel und den geheimen Schlüssel in eine Datei, damit Sie sie bei der Installation des Agenten verwenden können. Klicken Sie auf Schließen.

Erlauben Sie dem CloudWatch Agenten, eine Richtlinie zur Aufbewahrung von Protokollen festzulegen

Sie können den CloudWatch Agenten so konfigurieren, dass er die Aufbewahrungsrichtlinie für Protokollgruppen festlegt, an die er Protokollereignisse sendet. Wenn Sie dies tun, müssen Sie der IAM-Rolle oder dem Benutzer, den der Agent verwendet, die Berechtigung `logs:PutRetentionPolicy` gewähren. Der Agent verwendet eine IAM-Rolle für die Ausführung auf Amazon-EC2-Instances und nutzt für On-Premises-Server einen IAM-Benutzer.

Um der IAM-Rolle des CloudWatch Agenten die Berechtigung zu erteilen, Richtlinien zur Aufbewahrung von Protokollen festzulegen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Roles aus.
3. Geben Sie in das Suchfeld den Anfang des Namens der IAM-Rolle des CloudWatch Agenten ein. Das ist der Name, den Sie beim Erstellen der Rolle gewählt haben. Ein möglicher Name ist `CloudWatchAgentServerRole`.

   Wenn Sie die Rolle sehen, wählen Sie den Namen der Rolle aus.
4. Wählen Sie auf der Registerkarte Berechtigungen die Optionen Berechtigungen hinzufügen und dann Create Inline Policy (Inline-Richtlinie erstellen) aus.

5. Wählen Sie die Registerkarte JSON aus und kopieren Sie die folgende Richtlinie in das Feld. Dabei ersetzen Sie den Standard-JSON-Code im Feld:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "logs:PutRetentionPolicy",
      "Resource": "*"
    }
  ]
}
```

6. Wählen Sie Richtlinie prüfen.
7. Geben Sie für Name die Bezeichnung **`CloudWatchAgentPutLogsRetention`** oder etwas ähnliches ein und wählen Sie dann Create policy (Richtlinie erstellen) aus.

Um dem IAM-Benutzer des CloudWatch Agenten die Erlaubnis zu erteilen, Richtlinien zur Aufbewahrung von Protokollen festzulegen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Benutzer aus.
3. Geben Sie in das Suchfeld den Anfang des Namens des IAM-Benutzers des CloudWatch Agenten ein. Das ist der Name, den Sie beim Erstellen des Benutzers gewählt haben.

   Wenn Sie den Benutzer sehen, wählen Sie den Namen des Benutzers aus.
4. Wählen Sie auf der Registerkarte Permissions (Berechtigungen) die Option Add inline policy (Eingebundene Richtlinie hinzufügen).
5. Wählen Sie die Registerkarte JSON aus und kopieren Sie die folgende Richtlinie in das Feld. Dabei ersetzen Sie den Standard-JSON-Code im Feld:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "logs:PutRetentionPolicy",
```

```
      "Resource": "*"
    }
  ]
}
```

6. Wählen Sie Richtlinie prüfen.
7. Geben Sie für Name die Bezeichnung **`CloudWatchAgentPutLogsRetention`** oder etwas ähnliches ein und wählen Sie dann Create policy (Richtlinie erstellen) aus.

Installation und Ausführung des CloudWatch Agenten auf Ihren Servern

Folgen Sie nach dem Erstellen der Agentenkonfigurationsdatei und der IAM-Rolle oder des IAM-Benutzers diesen Schritten, um den Agenten auf Ihren Servern zu installieren und mithilfe dieser Konfiguration auszuführen. Fügen Sie dem Server zunächst eine IAM-Rolle oder einen IAM-Benutzer hinzu, die bzw. der den Agenten ausführt. Laden Sie anschließend auf diesem Server das Agentenpaket herunter und starten Sie es mit der erstellten Agentenkonfiguration.

Laden Sie das CloudWatch Agentenpaket über einen S3-Download-Link herunter

> **Note**
>
> Um den CloudWatch Agenten herunterzuladen, muss Ihre Verbindung TLS 1.2 oder höher verwenden.

Sie müssen den Agenten auf jedem Server installieren, auf dem Sie den Agenten ausführen.

Amazon Linux-AMIs

Der CloudWatch Agent ist als Paket in Amazon Linux 2023 und Amazon Linux 2 verfügbar. Wenn Sie dieses Betriebssystem verwenden, können Sie das Paket installieren, indem Sie den folgenden Befehl eingeben. Sie müssen außerdem sicherstellen, dass der IAM-Rolle, die der Instance zugewiesen ist, die CloudWatchAgentServerPolicyangehängt ist. Weitere Informationen finden Sie unter Erstellen Sie IAM-Rollen zur Verwendung mit dem CloudWatch Agenten auf Amazon EC2 EC2-Instances.

```
sudo yum install amazon-cloudwatch-agent
```

Alle Betriebssysteme

Auf allen unterstützten Betriebssystemen können Sie den CloudWatch Agenten über die Befehlszeile mit einem Amazon S3 S3-Download-Link herunterladen und installieren, wie in den folgenden Schritten beschrieben.

Für jeden Download-Link gibt es einen allgemeinen Link sowie Links für jede Region. Für Amazon Linux 2023 und Amazon Linux 2 und die x86-64-Architektur lauten beispielsweise drei der gültigen Download-Links:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-us-east-1.s3.us-east-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-eu-central-1.s3.eu-central-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

Architektur	Plattform	Download-Link	Link zur Signaturdatei
x86-64	Amazon Linux 2023 und Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/ .rpm amazon-cloudwatch-agent https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/amazon_linux/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/amazon_linux/amd64/latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	Centos	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/ amazon-cloudwatch-agent .rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/ amazon-cloudwatch-agent .rpm.sig

Architektur	Plattform	Download-Link	Link zur Signaturdatei
		https://amazoncloudwatch-agent — *Region .s3.region .amazonaws.com/centos/amd64/latest/* .rpm amazon-cloudwatch-agent	*https://amazoncloudwatch-agent — Region .s3.region .amazonaws.com/centos/amd64/latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3.region .amazonaws.com/redhat/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3.region .amazonaws.com/redhat/amd64/latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3.region .amazonaws.com/suse/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3.region .amazonaws.com/suse/amd64/latest/* .rpm.sig amazon-cloudwatch-agent

Architektur	Plattform	Download-Link	Link zur Signaturdatei
x86-64	Debian	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/debian/amd64/latest/* .deb amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/ amazon-cloudwatch-agent .deb.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/debian/amd64/latest/* .deb.sig amazon-cloudwatch-agent
x86-64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/ubuntu/amd64/latest/* .deb amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/ amazon-cloudwatch-agent .deb.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/ubuntu/amd64/latest/* .deb.sig amazon-cloudwatch-agent

Architektur	Plattform	Download-Link	Link zur Signaturdatei
x86-64	Oracle	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/oracle_linux/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/oracle_linux/amd64/latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	macOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/ amazon-cloudwatch-agent .pkg https://amazoncloudwatch-agent - *Region .s3. region .amazonaws.com/darwin/amd64/latest/* .pkg amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/ amazon-cloudwatch-agent .pkg.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/darwin/amd64/latest/* .pkg.sig amazon-cloudwatch-agent

Architektur	Plattform	Download-Link	Link zur Signaturdatei
x86-64	Windows	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/ amazon-cloudwatch-agent .msi https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/windows/amd64/latest/* .msi amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/ amazon-cloudwatch-agent .msi.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/windows/amd64/latest/* .msi.sig amazon-cloudwatch-agent
ARM64	Amazon Linux 2023 und Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/amazon_linux/arm64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/amazon_linux/arm64/latest/* .rpm.sig amazon-cloudwatch-agent

Architektur	Plattform	Download-Link	Link zur Signaturdatei
ARM64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/redhat/arm64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/redhat/arm64/latest/* .rpm.sig amazon-cloudwatch-agent
ARM64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/ubuntu/arm64/latest/* .deb amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/ amazon-cloudwatch-agent .deb.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/ubuntu/arm64/latest/* .deb.sig amazon-cloudwatch-agent

Architektur	Plattform	Download-Link	Link zur Signaturdatei
ARM64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/suse/arm64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/suse/arm64/latest/* .rpm.sig amazon-cloudwatch-agent

Um den CloudWatch Agenten über die Befehlszeile auf einer Amazon EC2 EC2-Instance zu installieren

1. Laden Sie den CloudWatch Agenten herunter. Geben Sie auf einem Linux-Server Folgendes ein. Bei *download-link* verwenden Sie den entsprechenden Download-Link aus der vorherigen Tabelle.

   ```
   wget download-link
   ```

   Bei einem Server mit Windows Server laden Sie die folgende Datei herunter:

   ```
   https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi
   ```

2. Nachdem Sie das Paket heruntergeladen haben, können Sie optional die Paketsignatur überprüfen. Weitere Informationen finden Sie unter Überprüfung der Signatur des Agentenpakets CloudWatch .

3. Installieren Sie das Paket . Wenn Sie ein RPM-Paket auf einen Linux-Server heruntergeladen haben, wechseln Sie in das Verzeichnis, das das Paket enthält, und geben Sie Folgendes ein:

   ```
   sudo rpm -U ./amazon-cloudwatch-agent.rpm
   ```

Wenn Sie ein DEB-Paket auf einen Linux-Server heruntergeladen haben, wechseln Sie in das Verzeichnis, das das Paket enthält, und geben Sie Folgendes ein:

```
sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
```

Wenn Sie ein MSI-Paket auf einem Server mit Windows Server heruntergeladen haben, wechseln Sie in das Verzeichnis, das das Paket enthält, und geben Sie Folgendes ein:

```
msiexec /i amazon-cloudwatch-agent.msi
```

Dieser Befehl funktioniert auch von innen heraus PowerShell. Weitere Informationen zu den MSI-Befehlsoptionen finden Sie unter Command-Line Options in der Microsoft Windows-Dokumentation.

(Installieren auf einer EC2-Instance) Verbinden einer IAM-Rolle

Damit der CloudWatch Agent Daten von der Instance senden kann, müssen Sie der Instance eine IAM-Rolle zuordnen. Die anzuhängende Rolle ist CloudWatchAgentServerRole. Sie hätten diese Rolle zuvor erstellen sollen. Weitere Informationen finden Sie unter Erstellen Sie IAM-Rollen und -Benutzer für die Verwendung mit dem Agenten CloudWatch .

Weitere Informationen zum Anhängen einer IAM-Rolle an eine Instance finden Sie unter Anhängen einer IAM-Rolle an eine Instance im Amazon-EC2-Benutzerhandbuch für Windows-Instances.

(Installation auf einem lokalen Server) Geben Sie die IAM-Anmeldeinformationen und die Region an AWS

Damit der CloudWatch Agent Daten von einem lokalen Server senden kann, müssen Sie den Zugriffsschlüssel und den geheimen Schlüssel des IAM-Benutzers angeben, den Sie zuvor erstellt haben. Weitere Informationen zum Erstellen dieses Benutzers finden Sie unter Erstellen Sie IAM-Rollen und -Benutzer für die Verwendung mit dem Agenten CloudWatch .

Sie müssen auch die AWS Region angeben, an die die Metriken gesendet werden sollen, indem Sie das `region` Feld im `[AmazonCloudWatchAgent]` Abschnitt der AWS Konfigurationsdatei verwenden, wie im folgenden Beispiel.

```
[profile AmazonCloudWatchAgent]
region = us-west-1
```

Im Folgenden finden Sie ein Beispiel für die Verwendung des `aws configure` Befehls, um ein benanntes Profil für den CloudWatch Agenten zu erstellen. Bei diesem Beispiel wird davon ausgegangen, dass Sie den Standardprofilnamen von `AmazonCloudWatchAgent` verwenden.

Um das AmazonCloudWatchAgent Profil für den CloudWatch Agenten zu erstellen

1. Falls Sie dies noch nicht getan haben, installieren Sie das AWS Command Line Interface auf dem Server. Weitere Informationen finden Sie unter Installieren der AWS CLI.
2. Geben Sie auf Linux-Servern den folgenden Befehl ein und befolgen Sie die Anweisungen:

   ```
   sudo aws configure --profile AmazonCloudWatchAgent
   ```

   Öffnen Sie Windows Server PowerShell als Administrator, geben Sie den folgenden Befehl ein und folgen Sie den Anweisungen.

   ```
   aws configure --profile AmazonCloudWatchAgent
   ```

Überprüfen des Internetzugangs

Ihre Amazon EC2 EC2-Instances müssen über ausgehenden Internetzugang verfügen, um Daten an CloudWatch oder CloudWatch Logs senden zu können. Weitere Informationen dazu, wie Sie den Internetzugang konfigurieren, finden Sie unter Internet-Gateways im Benutzerhandbuch zu Amazon VPC.

Folgende auf Ihrem Proxy zu konfigurierende Endpunkte und Ports sind möglich:

- Wenn Sie den Agenten zur Erfassung von Metriken verwenden, müssen Sie die CloudWatch Endpunkte für die entsprechenden Regionen zur Zulassungsliste hinzufügen. Diese Endpunkte sind unter CloudWatch Amazon-Endpunkte und Kontingente aufgeführt.
- Wenn Sie den Agenten zum Sammeln von Protokollen verwenden, müssen Sie die CloudWatch Logs-Endpunkte für die entsprechenden Regionen zur Zulassungsliste hinzufügen. Diese Endpunkte sind in Amazon CloudWatch Logs Endpoints and Quotas aufgeführt.
- Wenn Sie den Agenten mit dem Systems Manager installieren oder die Konfigurationsdatei mit Parameter Store speichern, müssen Sie die Systems-Manager-Endpunkte für die entsprechenden Regionen zur Allow-Liste hinzufügen. Diese Endpunkte sind unter AWS Systems Manager - Endpunkte und Kontingente aufgeführt.

(Optional) Ändern der gemeinsamen Konfiguration für Proxy- oder Regionsangaben

Der CloudWatch Agent enthält eine Konfigurationsdatei mit dem Namen. `common-config.toml` Sie können optional diese Datei verwenden, um Proxy- und Regionsinformationen anzugeben.

Auf einem Server, auf dem Linux ausgeführt wird, befindet sich diese Datei im Verzeichnis `/opt/aws/amazon-cloudwatch-agent/etc`. Auf einem Server, auf dem Windows Server ausgeführt wird, befindet sich diese Datei im Verzeichnis `C:\ProgramData\Amazon\AmazonCloudWatchAgent`.

> **Note**
>
> Wir empfehlen, dass Sie die `common-config.toml` Datei verwenden, um gemeinsam genutzte Konfigurationen und Anmeldeinformationen bereitzustellen, wenn Sie den CloudWatch Agenten in einem lokalen Modus ausführen. Sie kann auch nützlich sein, wenn Sie auf Amazon EC2 arbeiten und vorhandene Profile und Dateien mit gemeinsamen Anmeldeinformationen wiederverwenden möchten. Die Aktivierung über den `common-config.toml` hat den zusätzlichen Vorteil, dass, wenn Ihre Datei mit gemeinsam genutzten Anmeldeinformationen nach deren Ablauf mit erneuerten Anmeldeinformationen rotiert wird, die neuen Anmeldeinformationen automatisch vom Agenten übernommen werden, ohne dass ein Neustart erforderlich ist.

Die Datei `common-config.toml` hat standardmäßig folgenden Inhalt.

```
# This common-config is used to configure items used for both ssm and cloudwatch access


## Configuration for shared credential.
## Default credential strategy will be used if it is absent here:
##            Instance role is used for EC2 case by default.
##            AmazonCloudWatchAgent profile is used for the on-premises case by
 default.
# [credentials]
#    shared_credential_profile = "{profile_name}"
#    shared_credential_file= "{file_name}"

## Configuration for proxy.
## System-wide environment-variable will be read if it is absent here.
## i.e. HTTP_PROXY/http_proxy; HTTPS_PROXY/https_proxy; NO_PROXY/no_proxy
```

```
## Note: system-wide environment-variable is not accessible when using ssm run-command.
## Absent in both here and environment-variable means no proxy will be used.
# [proxy]
#    http_proxy = "{http_url}"
#    https_proxy = "{https_url}"
#    no_proxy = "{domain}"
```

Alle Zeilen sind anfangs auskommentiert. Zum Festlegen des Anmeldeinformationsprofils oder der Proxy-Einstellungen entfernen Sie # aus der Zeile und geben einen Wert an. Sie können diese Datei manuell oder mithilfe von `RunShellScript` Run Command in Systems Manager: bearbeiten:

- `shared_credential_profile`— Bei lokalen Servern gibt diese Zeile das Profil mit den IAM-Benutzeranmeldedaten an, an das Daten gesendet werden sollen. CloudWatch Wenn diese Zeile auskommentiert bleibt, wird `AmazonCloudWatchAgent` verwendet. Weitere Informationen zum Erstellen dieses Profils finden Sie unter (Installation auf einem lokalen Server) Geben Sie die IAM-Anmeldeinformationen und die Region an AWS.

  Auf einer EC2-Instance können Sie diese Zeile verwenden, damit der CloudWatch Agent Daten von dieser Instance CloudWatch in eine andere Region sendet. AWS Geben Sie hierzu ein benanntes Profil an, das ein Feld `region` zur Angabe der Zielregion enthält.

  Wenn Sie einen `shared_credential_profile` angeben, müssen Sie auch das # am Anfang der `[credentials]`-Zeile entfernen.
- `shared_credential_file` – Damit der Agent in einer nicht im Standardpfad abgelegten Datei nach Anmeldeinformationen sucht, müssen Sie den vollständigen Pfad und den Dateinamen hier angeben. Der Standardpfad ist unter Linux `/root/.aws` und unter Windows Server `C:\\Users\\Administrator\\.aws`.

  Das erste Beispiel unten zeigt die Syntax einer gültigen `shared_credential_file`-Zeile für Linux-Server, und das zweite Beispiel ist für Windows-Server gültig. Auf Windows Server müssen Sie die \-Zeichen mit einem Escape-Zeichen versehen.

  ```
  shared_credential_file= "/usr/username/credentials"
  ```

  ```
  shared_credential_file= "C:\\Documents and Settings\\username\\.aws\\credentials"
  ```

  Wenn Sie einen `shared_credential_file` angeben, müssen Sie auch das # am Anfang der `[credentials]`-Zeile entfernen.

- Proxy-Einstellungen – Falls Ihre Server HTTP- oder HTTPS-Proxys verwenden, um AWS -Services zu kontaktieren, geben Sie diese Proxys in den Feldern `http_proxy` und `https_proxy` an. Falls URLs vorhanden sind, die von Proxys ausgeschlossen werden sollen, geben Sie diese durch Kommas getrennt im Feld `no_proxy` an.

Starten Sie den CloudWatch Agenten über die Befehlszeile

Gehen Sie wie folgt vor, um den CloudWatch Agenten über die Befehlszeile auf einem Server zu starten.

Um den CloudWatch Agenten über die Befehlszeile auf einem Server zu starten

1. Kopieren Sie die gewünschte Agentenkonfigurationsdatei auf den Server, auf dem der Agent ausgeführt werden soll. Merken Sie sich den Pfadnamen, in den Sie die Datei kopiert haben.
2. Dieser Befehl `-a fetch-config` veranlasst den Agenten, die neueste Version der CloudWatch Agenten-Konfigurationsdatei zu laden, und `-s` startet den Agenten.

   Geben Sie einen der folgenden Befehle ein. *`configuration-file-path`*Ersetzen Sie durch den Pfad zur Agenten-Konfigurationsdatei. Diese Datei heißt `config.json`, wenn Sie sie mit dem Assistenten erstellt haben, und `amazon-cloudwatch-agent.json`, wenn Sie sie manuell erstellt haben.

   Geben Sie in einer EC2-Instance mit Linux den folgenden Befehl ein:

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
   config -m ec2 -s -c file:configuration-file-path
   ```

   Geben Sie auf einem On-Premises-Server mit Linux Folgendes ein:

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
   config -m onPremise -s -c file:configuration-file-path
   ```

   Geben Sie auf einer EC2-Instance, auf der Windows Server ausgeführt wird, Folgendes von der PowerShell Konsole aus ein:

   ```
   & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
   a fetch-config -m ec2 -s -c file:configuration-file-path
   ```

Geben Sie auf einem lokalen Server, auf dem Windows Server ausgeführt wird, Folgendes von der PowerShell Konsole aus ein:

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
a fetch-config -m onPremise -s -c file:configuration-file-path
```

Installieren Sie den CloudWatch Agenten mit AWS Systems Manager

Verwenden Sie die folgenden Themen, um den CloudWatch Agenten mit zu installieren und auszuführen. AWS Systems Manager

Themen

- Erstellen Sie IAM-Rollen und -Benutzer für die Verwendung mit dem Agenten CloudWatch
- Laden Sie den CloudWatch Agenten herunter und konfigurieren Sie ihn
- Den Agenten mithilfe Ihrer CloudWatch Agentenkonfiguration auf EC2-Instances installieren
- Installation des CloudWatch Agenten auf lokalen Servern

Erstellen Sie IAM-Rollen und -Benutzer für die Verwendung mit dem Agenten CloudWatch

Für den Zugriff auf AWS Ressourcen sind Berechtigungen erforderlich. Sie können IAM-Rollen und -Benutzer erstellen, die die Berechtigungen enthalten, die Sie für den CloudWatch Agenten zum Schreiben von Metriken CloudWatch und für die Kommunikation des CloudWatch Agenten mit Amazon EC2 und benötigen. AWS Systems Manager Sie verwenden IAM-Rollen auf der Amazon-EC2-Instance und IAM-Benutzer auf On-Premises-Servern.

Mit einer Rolle oder einem Benutzer kann der CloudWatch Agent auf einem Server installiert und Metriken an diesen gesendet werden. CloudWatch Die andere Rolle oder der andere Benutzer wird benötigt, um Ihre CloudWatch Agentenkonfiguration im Systems Manager Parameter Store zu speichern. Der Parameterspeicher ermöglicht es mehreren Servern, eine CloudWatch Agentenkonfiguration zu verwenden.

Die Möglichkeit zum Schreiben von Daten in Parameter Store ist eine umfassende und mächtige Berechtigung. Sie sollten sie nur dann verwenden, wenn es wirklich nötig ist. Zudem darf sie nicht mehreren Instances in Ihrer Bereitstellung zugewiesen werden. Wenn Sie Ihre CloudWatch Agentenkonfiguration im Parameter Store speichern, empfehlen wir Folgendes:

- Richten Sie eine Instance ein, in der Sie diese Konfiguration ausführen.
- Verwenden Sie die IAM-Rolle mit Berechtigungen zum Schreiben zu Parameter Store nur auf dieser Instance.
- Verwenden Sie die IAM-Rolle mit Schreibberechtigungen in den Parameter Store nur, während Sie mit der CloudWatch Agentenkonfigurationsdatei arbeiten und diese speichern.

> **Note**
>
> Wir haben die folgenden Verfahren kürzlich geändert, indem die neuen, von Amazon erstellten Richtlinien `CloudWatchAgentServerPolicy` und `CloudWatchAgentAdminPolicy` verwendet werden, anstatt dass Kunden diese Richtlinien selbst erstellen müssen. Damit die Agent-Konfigurationsdatei mithilfe dieser Richtlinien in Parameter Store geschrieben und dann von Parameter Store heruntergeladen werden kann, muss die Agent-Konfigurationsdatei einen Namen haben, der mit `AmazonCloudWatch-` beginnt. Wenn Sie über eine CloudWatch Agentenkonfigurationsdatei mit einem Dateinamen verfügen, der nicht mit beginnt`AmazonCloudWatch-`, können diese Richtlinien nicht verwendet werden, um die Datei in den Parameter Store zu schreiben oder die Datei aus dem Parameter Store herunterzuladen.

Erstellen Sie IAM-Rollen zur Verwendung mit dem CloudWatch Agenten auf Amazon EC2 EC2-Instances

Das erste Verfahren erstellt die IAM-Rolle, die Sie jeder Amazon EC2 EC2-Instance zuordnen müssen, auf der der CloudWatch Agent ausgeführt wird. Diese Rolle bietet Berechtigungen zum Lesen und Schreiben von Informationen aus der Instance. CloudWatch

Das zweite Verfahren erstellt die IAM-Rolle, die Sie der Amazon EC2 EC2-Instance zuordnen müssen, die zur Erstellung der CloudWatch Agenten-Konfigurationsdatei verwendet wird. Dieser Schritt ist notwendig, wenn Sie diese Datei in Systems Manager Parameter Store speichern möchten, damit andere Server sie verwenden können. Diese Rolle bietet neben den Berechtigungen zum Lesen und Schreiben von Informationen aus der Instance auch Berechtigungen zum Schreiben in den Parameter Store. CloudWatch Diese Rolle umfasst ausreichende Berechtigungen, um den CloudWatch Agenten auszuführen und in den Parameter Store zu schreiben.

> **Note**
>
> Parameter Store unterstützt Parameter in den Stufen „Standard“ und „Advanced“. Diese Parameterschichten haben nichts mit den Detailstufen Basic, Standard und Advanced zu tun, die in den vordefinierten Metriksätzen des CloudWatch Agenten verfügbar sind.

Um die IAM-Rolle zu erstellen, die für jeden Server zur Ausführung des CloudWatch Agenten erforderlich ist

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Rollen und dann Rolle erstellen.
3. Wählen Sie unter Select type of trusted entity (Typ der vertrauenswürdigen Entität auswählen) die Option AWS -Service aus.
4. Wählen Sie unter Häufige Anwendungsfälle die Option EC2 und dann Weiter: Berechtigungen.
5. Aktivieren Sie in der Liste der Richtlinien das Kontrollkästchen neben CloudWatchAgentServerPolicy. Verwenden Sie ggf. das Suchfeld, um die Richtlinie zu finden.
6. Um Systems Manager zur Installation oder Konfiguration des CloudWatch Agenten zu verwenden, aktivieren Sie das Kästchen neben AmazonSSM ManagedInstanceCore. Diese AWS verwaltete Richtlinie ermöglicht es einer Instanz, die Kernfunktionen des Systems Manager Manager-Service zu verwenden. Verwenden Sie ggf. das Suchfeld, um die Richtlinie zu finden. Diese Richtlinie ist nicht erforderlich, wenn Sie den Agenten nur über die Befehlszeile starten und konfigurieren.
7. Wählen Sie Weiter: Markierungen.
8. (Optional) Fügen Sie ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Rolle zu organisieren, zu verfolgen oder zu steuern und wählen Sie dann Next: Review (Weiter: Prüfen) aus.
9. Geben Sie unter Role name (Rollenname) einen Namen für Ihre neue Rolle, wie z. B. **`CloudWatchAgentServerRole`**, oder einen anderen von Ihnen bevorzugten Namen ein.
10. (Optional) Geben Sie im Feld Role description (Rollenbeschreibung) eine Beschreibung ein.
11. Bestätigen Sie, dass CloudWatchAgentServerPolicyund optional AmazonSSM neben ManagedInstanceCore Richtlinien angezeigt wird.
12. Wählen Sie Rolle erstellen aus.

Die Rolle wird jetzt erstellt.

In der folgenden Prozedur wird die IAM-Rolle erstellt, die auch in Parameter Store schreiben kann. Sie können diese Rolle verwenden, um die Agent-Konfigurationsdatei in Parameter Store zu speichern, sodass andere Server sie abrufen können.

Die Berechtigungen zum Schreiben in Parameter Store bieten umfassende Befugnisse. Diese Rolle darf nicht allen Ihren Servern zugewiesen werden. Sie darf nur von Administratoren verwendet werden. Nachdem Sie die Agentenkonfigurationsdatei erstellt und sie zu Parameter Store kopiert haben, sollten Sie diese Rolle von der Instance trennen und stattdessen `CloudWatchAgentServerRole` verwenden.

So erstellen Sie die IAM-Rolle für einen Administrator zum Schreiben in den Parameterspeicher

1. Melden Sie sich bei der an und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/ AWS Management Console .
2. Wählen Sie im Navigationsbereich Rollen und dann Rolle erstellen.
3. Wählen Sie unter Select type of trusted entity (Typ der vertrauenswürdigen Entität auswählen) die Option AWS -Service aus.
4. Wählen Sie für Choose the service that will use this role (Wählen Sie den Service aus, der diese Rolle verwendet) die Option EC2 und danach Next: Permissions (Nächster Schritt: Berechtigungen) aus.
5. Aktivieren Sie in der Liste der Richtlinien das Kontrollkästchen neben CloudWatchAgentAdminPolicy. Verwenden Sie ggf. das Suchfeld, um die Richtlinie zu finden.
6. Um Systems Manager zur Installation oder Konfiguration des CloudWatch Agenten zu verwenden, aktivieren Sie das Kästchen neben AmazonSSM ManagedInstanceCore. Diese AWS verwaltete Richtlinie ermöglicht es einer Instanz, die Kernfunktionen des Systems Manager Manager-Service zu verwenden. Verwenden Sie ggf. das Suchfeld, um die Richtlinie zu finden. Diese Richtlinie ist nicht erforderlich, wenn Sie den Agenten nur über die Befehlszeile starten und konfigurieren.
7. Wählen Sie Weiter: Markierungen.
8. (Optional) Fügen Sie ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Rolle zu organisieren, zu verfolgen oder zu steuern und wählen Sie dann Next: Review (Weiter: Prüfen) aus.

9. Geben Sie unter Role name (Rollenname) einen Namen für Ihre neue Rolle, wie z. B. **`CloudWatchAgentAdminRole`**, oder einen anderen von Ihnen bevorzugten Namen ein.
10. (Optional) Geben Sie im Feld Role description (Rollenbeschreibung) eine Beschreibung ein.
11. Bestätigen Sie, dass CloudWatchAgentAdminPolicyund optional AmazonSSM neben ManagedInstanceCore Richtlinien angezeigt wird.
12. Wählen Sie Rolle erstellen aus.

    Die Rolle wird jetzt erstellt.

Erstellen Sie IAM-Benutzer zur Verwendung mit dem CloudWatch Agenten auf lokalen Servern

Das erste Verfahren erstellt den IAM-Benutzer, den Sie zum Ausführen des Agenten benötigen. CloudWatch Dieser Benutzer erteilt Berechtigungen zum Senden von Daten an CloudWatch.

Das zweite Verfahren erstellt den IAM-Benutzer, den Sie beim Erstellen der CloudWatch Agent-Konfigurationsdatei verwenden können. Verwenden Sie dieses Verfahren, um in diese Datei in Systems Manager Parameter Store zu speichern, sodass andere Server sie verwenden können. Dieser Benutzer bietet zusätzlich zu den Berechtigungen zum Schreiben von Daten Berechtigungen zum Schreiben in den Parameter Store. CloudWatch

> **Note**
>
> Parameter Store unterstützt Parameter in den Stufen „Standard“ und „Advanced“. Diese Parameterschichten haben nichts mit den Detailebenen Basic, Standard und Advanced zu tun, die in den vordefinierten Metriksätzen von CloudWatch Agent verfügbar sind.

Um den IAM-Benutzer zu erstellen, in den der CloudWatch Agent Daten schreiben kann CloudWatch

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Users (Benutzer) und dann Add User (Benutzer hinzufügen) aus.
3. Geben Sie den Benutzernamen für den neuen Benutzer an.
4. Wählen Sie für Access type (Zugriffstyp) die Option Programmatic access (Programmgesteuerter Zugriff) und wählen Sie dann Next: Permissions (Nächster Schritt: Berechtigungen).

5. Wählen Sie unter Set permissions (Berechtigungen festlegen) die Option Attach existing policies directly (Vorhandene Richtlinien direkt anfügen) aus.
6. Aktivieren Sie in der Liste der Richtlinien das Kontrollkästchen neben CloudWatchAgentServerPolicy. Verwenden Sie ggf. das Suchfeld, um die Richtlinie zu finden.
7. Um Systems Manager zur Installation oder Konfiguration des CloudWatch Agenten zu verwenden, aktivieren Sie das Kästchen neben AmazonSSM ManagedInstanceCore. Diese AWS verwaltete Richtlinie ermöglicht es einer Instanz, die Kernfunktionen des Systems Manager Manager-Service zu verwenden. (Verwenden Sie bei Bedarf das Suchfeld, um die Richtlinie zu finden. Diese Richtlinie ist nicht erforderlich, wenn Sie den Agenten nur über die Befehlszeile starten und konfigurieren.)
8. Wählen Sie Weiter: Markierungen.
9. (Optional) Fügen Sie ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Rolle zu organisieren, zu verfolgen oder zu steuern und wählen Sie dann Next: Review (Weiter: Prüfen) aus.
10. Bestätigen Sie, dass die richtigen Richtlinien aufgelistet werden, und klicken Sie auf Create user (Benutzer erstellen).
11. Wählen Sie in der Zeile für den neuen Benutzer die Option Show (Anzeigen). Kopieren Sie den Zugriffsschlüssel und den geheimen Schlüssel in eine Datei, damit Sie sie bei der Installation des Agenten verwenden können. Klicken Sie auf Schließen.

In der folgenden Prozedur wird der IAM-Benutzer erstellt, der auch in Parameter Store schreiben kann. Sie müssen diesen IAM-Benutzer verwenden, wenn Sie vorhaben, die Agentenkonfigurationsdatei in Parameter Store zu speichern. Dieser IAM-Benutzer bietet Berechtigungen zum Schreiben in Parameter Store. Dieser Benutzer erteilt auch die Rechte zum Lesen und Schreiben von Informationen aus der Instanz CloudWatch. Die Berechtigungen zum Schreiben in Systems Manager Parameter Store bieten umfassende Befugnisse. Dieser IAM-Benutzer darf nicht allen Ihren Servern zugewiesen werden. Er darf nur von Administratoren verwendet werden. Verwenden Sie diesen IAM-Benutzer nur, wenn Sie die Agent-Konfigurationsdatei in Parameter Store speichern.

Um den IAM-Benutzer zu erstellen, müssen Sie die Konfigurationsdatei im Parameter Store speichern und Informationen an senden CloudWatch

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.

2. Wählen Sie im Navigationsbereich Users (Benutzer) und dann Add User (Benutzer hinzufügen) aus.
3. Geben Sie den Benutzernamen für den neuen Benutzer an.
4. Wählen Sie für Access type (Zugriffstyp) die Option Programmatic access (Programmgesteuerter Zugriff) und wählen Sie dann Next: Permissions (Nächster Schritt: Berechtigungen).
5. Wählen Sie unter Set permissions (Berechtigungen festlegen) die Option Attach existing policies directly (Vorhandene Richtlinien direkt anfügen) aus.
6. Aktivieren Sie in der Liste der Richtlinien das Kontrollkästchen neben CloudWatchAgentAdminPolicy. Verwenden Sie ggf. das Suchfeld, um die Richtlinie zu finden.
7. Um Systems Manager zur Installation oder Konfiguration des CloudWatch Agenten zu verwenden, aktivieren Sie das Kontrollkästchen neben AmazonSSM ManagedInstanceCore. Diese AWS verwaltete Richtlinie ermöglicht es einer Instanz, die Kernfunktionen des Systems Manager Manager-Service zu verwenden. (Verwenden Sie bei Bedarf das Suchfeld, um die Richtlinie zu finden. Diese Richtlinie ist nicht erforderlich, wenn Sie den Agenten nur über die Befehlszeile starten und konfigurieren.)
8. Wählen Sie Weiter: Markierungen.
9. (Optional) Fügen Sie ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Rolle zu organisieren, zu verfolgen oder zu steuern und wählen Sie dann Next: Review (Weiter: Prüfen) aus.
10. Bestätigen Sie, dass die richtigen Richtlinien aufgelistet werden, und klicken Sie auf Create user (Benutzer erstellen).
11. Wählen Sie in der Zeile für den neuen Benutzer die Option Show (Anzeigen). Kopieren Sie den Zugriffsschlüssel und den geheimen Schlüssel in eine Datei, damit Sie sie bei der Installation des Agenten verwenden können. Klicken Sie auf Schließen.

Laden Sie den CloudWatch Agenten herunter und konfigurieren Sie ihn

In diesem Abschnitt erfahren Sie, wie Sie Systems Manager zum Herunterladen des Agenten benutzen und eine Agentenkonfigurationsdatei erstellen. Bevor Sie Systems Manager verwenden können, um den Agenten herunterzuladen, müssen Sie sicherstellen, dass die Instance ordnungsgemäß für Systems Manager konfiguriert ist.

Installieren oder Aktualisieren von SSM-Agent

Auf einer Amazon EC2 EC2-Instance benötigt der CloudWatch Agent, dass auf der Instance Version 2.2.93.0 oder höher ausgeführt wird. Bevor Sie den CloudWatch Agenten installieren, aktualisieren oder installieren Sie den SSM-Agent auf der Instance, falls Sie dies noch nicht getan haben.

Informationen über das Installieren oder Aktualisieren des SSM Agent auf einer Instance, auf der Linux ausgeführt wird, finden Sie unter Installieren und Konfigurieren von SSM Agent auf Linux-Instances im AWS Systems Manager -Benutzerhandbuch.

Informationen über das Installieren oder Aktualisieren des SSM Agenten finden Sie unter Installieren und Konfigurieren des SSM-Agenten im AWS Systems Manager -Benutzerhandbuch.

(Optional) Überprüfen der Voraussetzungen für Systems Manager

Überprüfen des Internetzugangs

Ihre Amazon EC2 EC2-Instances müssen über ausgehenden Internetzugang verfügen, um Daten an CloudWatch oder CloudWatch Logs senden zu können. Weitere Informationen dazu, wie Sie den Internetzugang konfigurieren, finden Sie unter Internet-Gateways im Benutzerhandbuch zu Amazon VPC.

Folgende auf Ihrem Proxy zu konfigurierende Endpunkte und Ports sind möglich:

- Wenn Sie den Agenten zur Erfassung von Metriken verwenden, müssen Sie zulassen, dass Sie die CloudWatch Endpunkte für die entsprechenden Regionen auflisten. Diese Endpunkte sind bei Amazon CloudWatch in der Allgemeine Amazon Web Services-Referenzaufgeführt.
- Wenn Sie den Agenten zum Sammeln von Protokollen verwenden, müssen Sie die Liste der CloudWatch Logs-Endpunkte für die entsprechenden Regionen zulassen. Diese Endpunkte sind in Amazon CloudWatch Logs im Allgemeine Amazon Web Services-Referenzaufgeführt.
- Wenn Sie den Agenten mit Systems Manager installieren oder die Konfigurationsdatei mit Parameter Store speichern, müssen Sie die Systems-Manager-Endpunkte für die entsprechenden Regionen der Zulassungsliste hinzufügen. Diese Endpunkte sind unter AWS -Systems Manager im Allgemeine Amazon Web Services-Referenz aufgeführt.

Gehen Sie wie folgt vor, um das CloudWatch Agentenpaket mit Systems Manager herunterzuladen.

So laden Sie den CloudWatch Agenten mit Systems Manager herunter

1. Öffnen Sie die Systems Manager Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.

   –oder–

   Wenn die AWS Systems Manager Startseite geöffnet wird, scrollen Sie nach unten und wählen Sie Explore Run Command.
3. Wählen Sie Run Command (Befehl ausführen) aus.
4. Wählen Sie in der Liste der Befehlsdokumente die Option AWSPackageAWS-Configure aus.
5. Wählen Sie im Bereich Ziele die Instanz aus, auf der der CloudWatch Agent installiert werden soll. Wenn Sie eine bestimmte Instance nicht sehen, ist sie möglicherweise nicht als verwaltete Instance für die Verwendung mit Systems Manager konfiguriert. Weitere Informationen finden Sie im AWS Systems Manager Benutzerhandbuch unter Einrichtung AWS Systems Manager für Hybridumgebungen.
6. Klicken Sie in der Liste Action auf Install.
7. Geben Sie im Feld Name `AmazonCloudWatchAgent` ein.
8. Lassen Sie Version auf latest (aktuell) eingestellt, damit die neueste Version des Agenten installiert wird.
9. Wählen Sie Ausführen aus.
10. Wählen Sie optional in den Bereichen Targets and outputs (Ziele und Ausgaben) die Schaltfläche neben einem Instance-Namen aus und wählen Sie View output (Ausgabe anzeigen) aus. Systems Manager zeigt jetzt an, dass der Agent erfolgreich installiert wurde.

Erstellen und Ändern der Agentenkonfigurationsdatei

Nachdem Sie den CloudWatch Agenten heruntergeladen haben, müssen Sie die Konfigurationsdatei erstellen, bevor Sie den Agenten auf beliebigen Servern starten.

Wenn Sie die Agentenkonfigurationsdatei in Systems Manager Parameter Store speichern, müssen Sie eine EC2-Instance zum Speichern in Parameter Store verwenden. Darüber hinaus müssen Sie dieser Instance zunächst die IAM-Rolle `CloudWatchAgentAdminRole` anfügen. Weitere Informationen zum Anhängen von Rollen finden Sie unter Anhängen einer IAM-Rolle an eine Instance im Amazon-EC2-Benutzerhandbuch für Windows-Instances.

Weitere Informationen zum Erstellen der CloudWatch Agent-Konfigurationsdatei finden Sie unterErstellen Sie die CloudWatch Agent-Konfigurationsdatei.

Den Agenten mithilfe Ihrer CloudWatch Agentenkonfiguration auf EC2-Instances installieren

Nachdem Sie eine CloudWatch Agentenkonfiguration im Parameter Store gespeichert haben, können Sie sie verwenden, wenn Sie den Agenten auf anderen Servern installieren.

Themen

- Anfügen einer IAM-Rolle an die Instance
- Laden Sie das CloudWatch Agentenpaket auf eine Amazon EC2 EC2-Instance herunter
- (Optional) Ändern Sie die allgemeine Konfiguration und das benannte Profil für den CloudWatch Agenten
- Starten Sie den Agenten CloudWatch

Anfügen einer IAM-Rolle an die Instance

Sie müssen die CloudWatchAgentServerRoleIAM-Rolle an die EC2-Instance anhängen, um den CloudWatch Agenten auf der Instance ausführen zu können. Diese Rolle ermöglicht es dem CloudWatch Agenten, Aktionen auf der Instance durchzuführen. Sie hätten diese Rolle zuvor erstellen sollen. Weitere Informationen finden Sie unter Erstellen Sie IAM-Rollen und -Benutzer für die Verwendung mit dem Agenten CloudWatch .

Weitere Informationen finden Sie unter Anhängen einer IAM-Rolle an eine Instance im Amazon-EC2-Benutzerhandbuch für Windows-Instances.

Laden Sie das CloudWatch Agentenpaket auf eine Amazon EC2 EC2-Instance herunter

Sie müssen den Agenten auf jedem Server installieren, auf dem Sie den Agenten ausführen. Der CloudWatch Agent ist als Paket in Amazon Linux 2023 und Amazon Linux 2 verfügbar. Wenn Sie dieses Betriebssystem verwenden, können Sie das Paket installieren, indem Sie den folgenden Befehl eingeben. Sie müssen außerdem sicherstellen, dass der IAM-Rolle, die der Instance zugewiesen ist, die CloudWatchAgentServerPolicyangehängt ist. Weitere Informationen finden Sie unter Erstellen Sie IAM-Rollen zur Verwendung mit dem CloudWatch Agenten auf Amazon EC2 EC2-Instances.

```
sudo yum install amazon-cloudwatch-agent
```

Auf allen unterstützten Betriebssystemen können Sie das CloudWatch Agentenpaket entweder über Systems Manager Run Command oder über einen Amazon S3 S3-Download-Link herunterladen. Informationen zum Verwenden eines Amazon-S3-Download-Links finden Sie unter Laden Sie das CloudWatch Agentenpaket herunter.

> **Note**
>
> Wenn Sie den CloudWatch Agenten installieren oder aktualisieren, wird nur die Option Deinstallieren und Neuinstallieren unterstützt. Sie können die Option In-place update (Direkte Aktualisierung) nicht verwenden.

Laden Sie den CloudWatch Agenten mithilfe von Systems Manager auf eine Amazon EC2 EC2-Instance herunter

Bevor Sie Systems Manager zur Installation des CloudWatch Agenten verwenden können, müssen Sie sicherstellen, dass die Instanz korrekt für Systems Manager konfiguriert ist.

Installieren oder Aktualisieren von SSM-Agent

Auf einer Amazon EC2 EC2-Instance benötigt der CloudWatch Agent, dass auf der Instance Version 2.2.93.0 oder höher ausgeführt wird. Bevor Sie den CloudWatch Agenten installieren, aktualisieren oder installieren Sie den SSM-Agent auf der Instance, falls Sie dies noch nicht getan haben.

Informationen über das Installieren oder Aktualisieren des SSM Agent auf einer Instance, auf der Linux ausgeführt wird, finden Sie unter Installieren und Konfigurieren von SSM Agent auf Linux-Instances in AWS Systems Manager -Benutzerhandbuch.

Informationen zur Installation oder Aktualisierung von SSM Agent auf einer Instance mit Windows Server finden Sie unter Installieren und Konfigurieren des SSM-Agenten auf Windows-Instances im AWS Systems Manager -Benutzerhandbuch.

(Optional) Überprüfen der Voraussetzungen für Systems Manager

Bevor Sie Systems Manager Run Command zur Installation und Konfiguration des CloudWatch Agenten verwenden, stellen Sie sicher, dass Ihre Instances die Mindestanforderungen von Systems Manager erfüllen. Weitere Informationen finden Sie unter Einrichten von AWS Systems Manager im Benutzerhandbuch für AWS Systems Manager .

Überprüfen des Internetzugangs

Ihre Amazon EC2 EC2-Instances müssen über ausgehenden Internetzugang verfügen, um Daten an CloudWatch oder CloudWatch Logs senden zu können. Weitere Informationen dazu, wie Sie den Internetzugang konfigurieren, finden Sie unter Internet-Gateways im Benutzerhandbuch zu Amazon VPC.

Laden Sie das CloudWatch Agentenpaket herunter

Systems Manager Run Command ermöglicht Ihnen die bedarfsgerechte Verwaltung der Konfiguration Ihrer Instances. Sie geben ein Systems-Manager-Dokument und Parameter an und führen Sie den Befehl auf einer oder mehreren Instances aus. Der SSM-Agent auf der Instance verarbeitet den Befehl und konfiguriert die Instance wie angegeben.

Um den CloudWatch Agenten mit Run Command herunterzuladen

1. Öffnen Sie die Systems Manager Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.

   –oder–

   Wenn die AWS Systems Manager Startseite geöffnet wird, scrollen Sie nach unten und wählen Sie Explore Run Command.
3. Wählen Sie Run Command (Befehl ausführen) aus.
4. Wählen Sie in der Liste der Befehlsdokumente die Option AWSPackageAWS-Configure aus.
5. Wählen Sie im Bereich Ziele die Instanz aus, auf der der Agent installiert werden soll. CloudWatch Wenn Sie eine bestimmte Instance nicht sehen, ist sie möglicherweise nicht für Run Command konfiguriert. Weitere Informationen erhalten Sie unter Einrichten von AWS Systems Manager für Hybridumgebungen im AWS Systems Manager -Benutzerhandbuch.
6. Klicken Sie in der Liste Action auf Install.
7. Geben Sie im Feld Name (Name) `AmazonCloudWatchAgent` ein.
8. Lassen Sie Version auf latest (aktuell) eingestellt, damit die neueste Version des Agenten installiert wird.
9. Wählen Sie Ausführen aus.

10. Wählen Sie optional in den Bereichen Targets and outputs (Ziele und Ausgaben) die Schaltfläche neben einem Instance-Namen aus und wählen Sie View output (Ausgabe anzeigen) aus. Systems Manager zeigt jetzt an, dass der Agent erfolgreich installiert wurde.

(Optional) Ändern Sie die allgemeine Konfiguration und das benannte Profil für den CloudWatch Agenten

Der CloudWatch Agent enthält eine Konfigurationsdatei mit dem Namen`common-config.toml`. Sie können diese Datei verwenden, um optional Proxy- und Regionsinformationen anzugeben.

Auf einem Server, auf dem Linux ausgeführt wird, befindet sich diese Datei im Verzeichnis `/opt/aws/amazon-cloudwatch-agent/etc`. Auf einem Server, auf dem Windows Server ausgeführt wird, befindet sich diese Datei im Verzeichnis `C:\ProgramData\Amazon\AmazonCloudWatchAgent`.

`common-config.toml` lautet standardmäßig folgendermaßen:

```
# This common-config is used to configure items used for both ssm and cloudwatch access


## Configuration for shared credential.
## Default credential strategy will be used if it is absent here:
##            Instance role is used for EC2 case by default.
##            AmazonCloudWatchAgent profile is used for onPremise case by default.
# [credentials]
#    shared_credential_profile = "{profile_name}"
#    shared_credential_file= "{file_name}"

## Configuration for proxy.
## System-wide environment-variable will be read if it is absent here.
## i.e. HTTP_PROXY/http_proxy; HTTPS_PROXY/https_proxy; NO_PROXY/no_proxy
## Note: system-wide environment-variable is not accessible when using ssm run-command.
## Absent in both here and environment-variable means no proxy will be used.
# [proxy]
#    http_proxy = "{http_url}"
#    https_proxy = "{https_url}"
#    no_proxy = "{domain}"
```

Alle Zeilen sind anfangs auskommentiert. Zum Festlegen des Anmeldeinformationsprofils oder der Proxy-Einstellungen entfernen Sie # aus der Zeile und geben einen Wert an. Sie können diese Datei manuell oder mithilfe von `RunShellScript` Run Command in Systems Manager: bearbeiten:

- `shared_credential_profile`— Für lokale Server gibt diese Zeile das Profil mit den IAM-Benutzeranmeldeinformationen an, an das Daten gesendet werden sollen. CloudWatch Wenn diese Zeile auskommentiert bleibt, wird `AmazonCloudWatchAgent` verwendet.

  Auf einer EC2-Instance können Sie diese Zeile verwenden, damit der CloudWatch Agent Daten von dieser Instance CloudWatch in eine andere Region sendet. AWS Geben Sie hierzu ein benanntes Profil an, das ein Feld `region` zur Angabe der Zielregion enthält.

  Wenn Sie einen `shared_credential_profile` angeben, müssen Sie auch das # am Anfang der `[credentials]`-Zeile entfernen.
- `shared_credential_file` – Damit der Agent in einer nicht im Standardpfad abgelegten Datei nach Anmeldeinformationen sucht, müssen Sie den vollständigen Pfad und den Dateinamen hier angeben. Der Standardpfad ist unter Linux `/root/.aws` und unter Windows Server `C:\\Users\\Administrator\\.aws`.

  Das erste Beispiel unten zeigt die Syntax einer gültigen `shared_credential_file`-Zeile für Linux-Server, und das zweite Beispiel ist für Windows-Server gültig. Auf Windows Server müssen Sie die \-Zeichen mit einem Escape-Zeichen versehen.

  ```
  shared_credential_file= "/usr/username/credentials"
  ```

  ```
  shared_credential_file= "C:\\Documents and Settings\\username\\.aws\\credentials"
  ```

  Wenn Sie einen `shared_credential_file` angeben, müssen Sie auch das # am Anfang der `[credentials]`-Zeile entfernen.
- Proxy-Einstellungen – Falls Ihre Server HTTP- oder HTTPS-Proxys verwenden, um AWS -Services zu kontaktieren, geben Sie diese Proxys in den Feldern `http_proxy` und `https_proxy` an. Falls URLs vorhanden sind, die von Proxys ausgeschlossen werden sollen, geben Sie diese durch Kommas getrennt im Feld `no_proxy` an.

Starten Sie den Agenten CloudWatch

Sie können den Agenten mit dem Systems Manager Run Command oder der Befehlszeile starten.

Starten Sie den CloudWatch Agenten mit Systems Manager Run Command

Führen Sie diese Schritte aus, um den Agenten mit dem Systems Manager Run Command zu starten.

Um den CloudWatch Agenten mit Run Command zu starten

1. Öffnen Sie die Systems Manager Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.

   –oder–

   Wenn die AWS Systems Manager Startseite geöffnet wird, scrollen Sie nach unten und wählen Sie Explore Run Command.
3. Wählen Sie Run Command (Befehl ausführen) aus.
4. Wählen Sie in der Command-Dokumentliste die Option AmazonCloudWatch- ManageAgent.
5. Wählen Sie im Bereich Ziele die Instanz aus, auf der Sie den CloudWatch Agenten installiert haben.
6. Klicken Sie in der Liste Action auf Configure.
7. Klicken Sie in der Liste Optional Configuration Source auf ssm.
8. Geben Sie in das Feld Optionaler Konfigurationsstandort den Namen des Systems-Manager-Parameternamens der Agenten-Konfigurationsdatei ein, die Sie erstellt und im Systems-Manager-Parameter-Speicher gespeichert haben, wie in Erstellen Sie die CloudWatch Agent-Konfigurationsdatei erläutert.
9. Klicken Sie in der Liste Optional Restart auf yes, um den Agent zu starten, nachdem Sie diese Schritte abgeschlossen haben.
10. Wählen Sie Ausführen aus.
11. Wählen Sie optional in den Bereichen Targets and outputs (Ziele und Ausgaben) die Schaltfläche neben einem Instance-Namen aus und wählen Sie View output (Ausgabe anzeigen) aus. Systems Manager zeigt jetzt an, dass der Agent erfolgreich gestartet wurde.

Starten Sie den CloudWatch Agenten auf einer Amazon EC2 EC2-Instance über die Befehlszeile

Gehen Sie wie folgt vor, um den CloudWatch Agenten über die Befehlszeile auf einer Amazon EC2 EC2-Instance zu installieren.

So verwenden Sie die Befehlszeile, um den CloudWatch Agenten auf einer Amazon EC2 EC2-Instance zu starten

- Dieser Befehl `-a fetch-config` veranlasst den Agenten, die neueste Version der CloudWatch Agenten-Konfigurationsdatei zu laden, und `-s` startet den Agenten.

  Linux und macOS: Wenn Sie die Konfigurationsdatei in Systems Manager Parameter Store gespeichert haben, geben Sie Folgendes ein:

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
  config -m ec2 -s -c ssm:configuration-parameter-store-name
  ```

  Linux und macOS: Wenn Sie die Konfigurationsdatei auf dem lokalen Computer gespeichert haben, geben Sie den folgenden Befehl ein: *configuration-file-path*Ersetzen Sie durch den Pfad zur Agenten-Konfigurationsdatei. Diese Datei heißt `config.json`, wenn Sie sie mit dem Assistenten erstellt haben, und `amazon-cloudwatch-agent.json`, wenn Sie sie manuell erstellt haben.

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
  config -m ec2 -s -c file:configuration-file-path
  ```

  Windows Server: Wenn Sie die Agent-Konfigurationsdatei im Systems Manager Parameter Store gespeichert haben, geben Sie in der PowerShell Konsole Folgendes ein:

  ```
  & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
  a fetch-config -m ec2 -s -c ssm:configuration-parameter-store-name
  ```

  Windows Server: Wenn Sie die Agent-Konfigurationsdatei auf dem lokalen Computer gespeichert haben, geben Sie in der PowerShell Konsole Folgendes ein:

  ```
  & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
   -a fetch-config -m ec2 -s -c file:"C:\Program Files\Amazon\AmazonCloudWatchAgent
  \config.json"
  ```

Installation des CloudWatch Agenten auf lokalen Servern

Wenn Sie den CloudWatch Agenten auf einen Computer heruntergeladen und die gewünschte Agentenkonfigurationsdatei erstellt haben, können Sie diese Konfigurationsdatei verwenden, um den Agenten auf anderen lokalen Servern zu installieren.

Laden Sie den CloudWatch Agenten auf einen lokalen Server herunter

Sie können das CloudWatch Agentenpaket entweder über Systems Manager Run Command oder über einen Amazon S3 S3-Download-Link herunterladen. Informationen zum Verwenden eines Amazon-S3-Download-Links finden Sie unter Laden Sie das CloudWatch Agentenpaket herunter.

Download mithilfe des Systems Manager

Um Systems Manager Run Command verwenden zu können, müssen Sie Ihren On-Premises-Server bei Amazon EC2 Systems Manager registrieren. Weitere Informationen erhalten Sie unter Einrichten von Systems Manager in Hybridumgebungen im AWS Systems Manager -Benutzerhandbuch.

Wenn Sie Ihren Server bereits registriert haben, aktualisieren Sie SSM Agent auf die neueste Version.

Weitere Informationen zum Aktualisieren von SSM Agent auf einem Server, auf dem Linux ausgeführt wird, finden Sie unter Installieren von SSM Agent für eine Hybridumgebung (Linux) im AWS Systems Manager -Benutzerhandbuch.

Weitere Informationen zum Aktualisieren von SSM Agent auf einem Server, auf dem Windows Server ausgeführt wird, finden Sie unter Installieren von SSM Agent für eine Hybridumgebung (Windows) im AWS Systems Manager -Benutzerhandbuch.

Um den SSM-Agenten zum Herunterladen des CloudWatch Agentenpakets auf einen lokalen Server zu verwenden

1. Öffnen Sie die Systems Manager Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.

   –oder–

   Wenn die AWS Systems Manager Startseite geöffnet wird, scrollen Sie nach unten und wählen Sie Explore Run Command.

3. Wählen Sie Run Command (Befehl ausführen) aus.
4. Wählen Sie in der Liste der Befehlsdokumente die Schaltfläche neben AWSPackageAWS-Configure aus.
5. Wählen Sie im Bereich Ziele den Server aus, auf dem der CloudWatch Agent installiert werden soll. Wenn Sie einen bestimmten Server nicht sehen, ist er möglicherweise nicht für Run Command konfiguriert. Weitere Informationen erhalten Sie unter Einrichten von AWS Systems Manager für Hybridumgebungen im AWS Systems Manager -Benutzerhandbuch.
6. Klicken Sie in der Liste Action auf Install.
7. Geben Sie im Feld Name (Name) *AmazonCloudWatchAgent* ein.
8. Lassen Sie Version leer, damit die aktuelle Version des Agenten installiert wird.
9. Wählen Sie Ausführen aus.

   Das Agent-Paket wird heruntergeladen und die nächsten Schritte sind die Konfiguration und der Start.

(Installation auf einem lokalen Server) Geben Sie die IAM-Anmeldeinformationen und die Region an AWS

Damit der CloudWatch Agent Daten von einem lokalen Server senden kann, müssen Sie den Zugriffsschlüssel und den geheimen Schlüssel des IAM-Benutzers angeben, den Sie zuvor erstellt haben. Weitere Informationen zum Erstellen dieses Benutzers finden Sie unter Erstellen Sie IAM-Rollen und -Benutzer für die Verwendung mit dem Agenten CloudWatch .

In diesem Feld müssen Sie auch die AWS Region angeben, an die die Messwerte gesendet werden sollen. `region`

Im Folgenden wird ein Beispiel für diese Datei gezeigt.

```
[AmazonCloudWatchAgent]
aws_access_key_id=my_access_key
aws_secret_access_key=my_secret_key
region = us-west-1
```

Verwenden Sie anstelle von *my_access_key* und *my_secret_key* die Schlüssel des IAM-Benutzers, der keine Schreibrechte für Systems Manager Parameter Store hat. Weitere Informationen zu den IAM-Benutzern, die für den CloudWatch Agenten benötigt werden, finden Sie unterErstellen Sie IAM-Benutzer zur Verwendung mit dem CloudWatch Agenten auf lokalen Servern.

Wenn Sie das Profil `AmazonCloudWatchAgent` nennen, müssen Sie nichts weiter tun. Optional können Sie einen anderen Namen vergeben und diesen Namen als Wert für `shared_credential_profile` in der Datei `common-config.toml` angeben (siehe folgender Abschnitt).

Im Folgenden finden Sie ein Beispiel für die Verwendung des aws configure Befehls zum Erstellen eines benannten Profils für den CloudWatch Agenten. Bei diesem Beispiel wird davon ausgegangen, dass Sie den Standardprofilnamen `AmazonCloudWatchAgent` verwenden.

Um das AmazonCloudWatchAgent Profil für den CloudWatch Agenten zu erstellen

1. Falls Sie dies noch nicht getan haben, installieren Sie das AWS Command Line Interface auf dem Server. Weitere Informationen finden Sie unter Installieren der AWS CLI.
2. Geben Sie auf Linux-Servern den folgenden Befehl ein und befolgen Sie die Anweisungen:

   ```
   sudo aws configure --profile AmazonCloudWatchAgent
   ```

   Öffnen Sie Windows Server PowerShell als Administrator, geben Sie den folgenden Befehl ein und folgen Sie den Anweisungen.

   ```
   aws configure --profile AmazonCloudWatchAgent
   ```

(Optional) Ändern der allgemeinen Konfiguration und des benannten Profils für den Agenten CloudWatch

Der CloudWatch Agent enthält eine Konfigurationsdatei mit dem Namen`common-config.toml`. Sie können optional diese Datei verwenden, um Proxy- und Regionsinformationen anzugeben.

Auf einem Server, auf dem Linux ausgeführt wird, befindet sich diese Datei im Verzeichnis `/opt/aws/amazon-cloudwatch-agent/etc`. Auf einem Server, auf dem Windows Server ausgeführt wird, befindet sich diese Datei im Verzeichnis `C:\ProgramData\Amazon\AmazonCloudWatchAgent`.

`common-config.toml` lautet standardmäßig folgendermaßen:

```
# This common-config is used to configure items used for both ssm and cloudwatch access
```

```
## Configuration for shared credential.
## Default credential strategy will be used if it is absent here:
##            Instance role is used for EC2 case by default.
##            AmazonCloudWatchAgent profile is used for onPremise case by default.
# [credentials]
#    shared_credential_profile = "{profile_name}"
#    shared_credential_file= "{file_name}"

## Configuration for proxy.
## System-wide environment-variable will be read if it is absent here.
## i.e. HTTP_PROXY/http_proxy; HTTPS_PROXY/https_proxy; NO_PROXY/no_proxy
## Note: system-wide environment-variable is not accessible when using ssm run-command.
## Absent in both here and environment-variable means no proxy will be used.
# [proxy]
#    http_proxy = "{http_url}"
#    https_proxy = "{https_url}"
#    no_proxy = "{domain}"
```

Alle Zeilen sind anfangs auskommentiert. Zum Festlegen des Anmeldeinformationsprofils oder der Proxy-Einstellungen entfernen Sie # aus der Zeile und geben einen Wert an. Sie können diese Datei manuell oder mithilfe von `RunShellScript` Run Command in Systems Manager: bearbeiten:

- `shared_credential_profile`— Für lokale Server gibt diese Zeile das Profil mit den IAM-Benutzeranmeldeinformationen an, an das Daten gesendet werden sollen. CloudWatch Wenn diese Zeile auskommentiert bleibt, wird `AmazonCloudWatchAgent` verwendet. Weitere Informationen zum Erstellen dieses Profils finden Sie unter (Installation auf einem lokalen Server) Geben Sie die IAM-Anmeldeinformationen und die Region an AWS.

  Auf einer EC2-Instance können Sie diese Zeile verwenden, damit der CloudWatch Agent Daten von dieser Instance CloudWatch in eine andere Region sendet. AWS Geben Sie hierzu ein benanntes Profil an, das ein Feld `region` zur Angabe der Zielregion enthält.

  Wenn Sie einen `shared_credential_profile` angeben, müssen Sie auch das # am Anfang der `[credentials]`-Zeile entfernen.
- `shared_credential_file` – Damit der Agent in einer nicht im Standardpfad abgelegten Datei nach Anmeldeinformationen sucht, müssen Sie den vollständigen Pfad und den Dateinamen hier angeben. Der Standardpfad ist unter Linux `/root/.aws` und unter Windows Server `C:\\Users\\Administrator\\.aws`.

Das erste Beispiel unten zeigt die Syntax einer gültigen `shared_credential_file`-Zeile für Linux-Server, und das zweite Beispiel ist für Windows-Server gültig. Auf Windows Server müssen Sie die \-Zeichen mit einem Escape-Zeichen versehen.

```
shared_credential_file= "/usr/username/credentials"
```

```
shared_credential_file= "C:\\Documents and Settings\\username\\.aws\\credentials"
```

Wenn Sie einen `shared_credential_file` angeben, müssen Sie auch das # am Anfang der `[credentials]`-Zeile entfernen.

- Proxy-Einstellungen – Falls Ihre Server HTTP- oder HTTPS-Proxys verwenden, um AWS -Services zu kontaktieren, geben Sie diese Proxys in den Feldern `http_proxy` und `https_proxy` an. Falls URLs vorhanden sind, die von Proxys ausgeschlossen werden sollen, geben Sie diese durch Kommas getrennt im Feld `no_proxy` an.

Starten des CloudWatch-Agenten

Sie können den CloudWatch Agenten entweder mit Systems Manager Run Command oder der Befehlszeile starten.

So verwenden Sie den SSM-Agent, um den CloudWatch Agenten auf einem lokalen Server zu starten

1. Öffnen Sie die Systems Manager Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.

   –oder–

   Wenn die AWS Systems Manager Startseite geöffnet wird, scrollen Sie nach unten und wählen Sie Explore Run Command.
3. Wählen Sie Run Command (Befehl ausführen) aus.
4. Wählen Sie in der Liste der Befehlsdokumente die Schaltfläche neben AmazonCloudWatch-ausManageAgent.
5. Wählen Sie im Bereich Targets die Instance aus, auf der Sie den Agent installiert haben.
6. Klicken Sie in der Liste Action auf Configure.

7. Wählen Sie in der Liste Mode onPremise.
8. Geben Sie im Feld Optional Configuration Location (Optionaler Konfigurationsstandort) den Namen der Agentenkonfigurationsdatei ein, die Sie mit dem Assistenten erstellt und in Parameter Store gespeichert haben.
9. Wählen Sie Ausführen aus.

   Der Agent beginnt mit der Konfiguration, die Sie in der Konfigurationsdatei angegeben haben.

Um den CloudWatch Agenten über die Befehlszeile auf einem lokalen Server zu starten

- Dieser Befehl `-a fetch-config` veranlasst den Agenten, die neueste Version der CloudWatch Agenten-Konfigurationsdatei zu laden, und `-s` startet den Agenten.

  Linux: Wenn Sie die Konfigurationsdatei in Systems Manager Parameter Store gespeichert haben, geben Sie Folgendes ein:

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
  config -m onPremise -s -c ssm:configuration-parameter-store-name
  ```

  Linux: Wenn Sie die Konfigurationsdatei auf dem lokalen Computer gespeichert haben, geben Sie den folgenden Befehl ein: *configuration-file-path*Ersetzen Sie durch den Pfad zur Agenten-Konfigurationsdatei. Diese Datei heißt `config.json`, wenn Sie sie mit dem Assistenten erstellt haben, und `amazon-cloudwatch-agent.json`, wenn Sie sie manuell erstellt haben.

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
  config -m onPremise -s -c file:configuration-file-path
  ```

  Windows Server: Wenn Sie die Agent-Konfigurationsdatei im Systems Manager Parameter Store gespeichert haben, geben Sie in der PowerShell Konsole Folgendes ein:

  ```
  & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
  a fetch-config -m onPremise -s -c ssm:configuration-parameter-store-name
  ```

  Windows Server: Wenn Sie die Agent-Konfigurationsdatei auf dem lokalen Computer gespeichert haben, geben Sie in der PowerShell Konsole Folgendes ein. *configuration-file-path*Ersetzen Sie es durch den Pfad zur Agenten-Konfigurationsdatei. Diese Datei heißt

`config.json`, wenn Sie sie mit dem Assistenten erstellt haben, und `amazon-cloudwatch-agent.json`, wenn Sie sie manuell erstellt haben.

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
a fetch-config -m onPremise -s -c file:configuration-file-path
```

Installieren Sie den CloudWatch Agenten auf neuen Instanzen mit AWS CloudFormation

Amazon hat mehrere AWS CloudFormation Vorlagen hochgeladen, GitHub um Ihnen bei der Installation und Aktualisierung des CloudWatch Agenten auf neuen Amazon EC2 EC2-Instances zu helfen. Weitere Informationen zur Verwendung finden Sie AWS CloudFormation unter Was ist AWS CloudFormation? .

Der Speicherort der Vorlage lautet Deploy the Amazon CloudWatch Agent to EC2 Instances using AWS CloudFormation. Dort finden Sie die beiden Verzeichnisse `inline` und `ssm`. Jedes dieser Verzeichnisse enthält Vorlagen für Linux- und Windows-Instances.

- Bei den Vorlagen im `inline` Verzeichnis ist die CloudWatch Agentenkonfiguration in die AWS CloudFormation Vorlage eingebettet. Standardmäßig erfassen die Linux-Vorlagen die Metriken `mem_used_percent` und `swap_used_percent`, die Windows-Vorlagen dagegen `Memory % Committed Bytes In Use` und `Paging File % Usage`.

  Sie können diese Vorlagen ändern, um andere Metriken zu erfassen, indem Sie den folgenden Abschnitt der Vorlage ändern. Das folgende Beispiel stammt aus der Vorlage für Linux-Server. Befolgen Sie das Format und die Syntax der Agentenkonfigurationsdatei, um diese Änderungen vorzunehmen. Weitere Informationen finden Sie unter Erstellen oder bearbeiten Sie die CloudWatch Agenten-Konfigurationsdatei manuell.

```
{
   "metrics":{
      "append_dimensions":{
         "AutoScalingGroupName":"${!aws:AutoScalingGroupName}",
         "ImageId":"${!aws:ImageId}",
         "InstanceId":"${!aws:InstanceId}",
         "InstanceType":"${!aws:InstanceType}"
      },
      "metrics_collected":{
```

```
        "mem":{
           "measurement":[
              "mem_used_percent"
           ]
        },
        "swap":{
           "measurement":[
              "swap_used_percent"
           ]
        }
     }
  }
}
```

> **Note**
>
> In den Inline-Vorlagen müssen alle Platzhaltervariablen ein Ausrufezeichen (!) als Escape-Zeichen vor sich haben. Dies sehen Sie in der Beispielvorlage. Wenn Sie weitere Platzhaltervariablen hinzufügen, achten Sie darauf, dass Sie vor dem Namen ein Ausrufezeichen hinzufügen.

- Die Vorlagen im Verzeichnis `ssm` laden eine Agentenkonfigurationsdatei aus Parameter Store. Um diese Vorlagen verwenden zu können, müssen Sie zunächst eine Konfigurationsdatei erstellen und diese in Parameter Store hochladen. Sie stellen dann den Parameter-Store-Namen der Datei in der Vorlage bereit. Sie können die Konfigurationsdatei manuell oder mit Hilfe des Assistenten erstellen. Weitere Informationen finden Sie unter Erstellen Sie die CloudWatch Agent-Konfigurationsdatei.

Sie können beide Arten von Vorlagen für die Installation des CloudWatch Agenten und für die Aktualisierung der Agentenkonfiguration verwenden.

Tutorial: Installieren und konfigurieren Sie den CloudWatch Agenten mithilfe einer AWS CloudFormation Inline-Vorlage

In diesem Tutorial erfahren Sie AWS CloudFormation , wie Sie den CloudWatch Agenten auf einer neuen Amazon EC2 EC2-Instance installieren. Dieses Tutorial wird auf einer neuen Instance installiert, die Amazon Linux 2 mit den Inline-Vorlagen ausführt, für die weder die JSON-Konfigurationsdatei noch Parameter Store benötigt wird. Die Inline-Vorlage enthält die Agent-Konfiguration in der Vorlage. In diesem Tutorial verwenden Sie die in der Vorlage enthaltene Standardagentenkonfiguration.

Nach der Vorgehensweise zur Installation des Agenten fährt das Tutorial mit der Aktualisierung des Agenten fort.

Wird verwendet AWS CloudFormation , um den CloudWatch Agenten auf einer neuen Instance zu installieren

1. Laden Sie die Vorlage von herunter GitHub. Laden Sie in diesem Tutorial die Inline-Vorlage für Amazon Linux 2 wie folgt herunter:

   ```
   curl -O https://raw.githubusercontent.com/awslabs/aws-cloudformation-templates/
   master/aws/solutions/AmazonCloudWatchAgent/inline/amazon_linux.template
   ```

2. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
3. Wählen Sie Stack erstellen aus.
4. Wählen Sie für Choose a template (Vorlage auswählen) Upload a template to Amazon S3 (Vorlage auf Amazon S3 hochladen), wählen Sie die heruntergeladene Vorlage aus und klicken Sie auf Next (Weiter).
5. Geben Sie auf der Seite Specify Details (Details angeben) die folgenden Parameter ein und wählen Sie Next (Weiter) aus:

   - Stack-Name: Wählen Sie einen Stack-Namen für Ihren AWS CloudFormation Stack.
   - iamRole: Wählen Sie eine IAM-Rolle aus, die berechtigt ist, CloudWatch Metriken, Logs und Traces zu schreiben. Weitere Informationen finden Sie unter Erstellen Sie IAM-Rollen zur Verwendung mit dem CloudWatch Agenten auf Amazon EC2 EC2-Instances.
   - InstanceAMI: Wählen Sie ein AMI, das in der Region gültig ist, in der Sie Ihren Stack starten werden.
   - InstanceType: Wählen Sie einen gültigen Instance-Typ.
   - KeyName: Um den SSH-Zugriff auf die neue Instance zu aktivieren, wählen Sie ein vorhandenes Amazon EC2 EC2-Schlüsselpaar aus. Wenn Sie noch kein Amazon-EC2-Schlüsselpaar haben, können Sie eines in der AWS Management Console erstellen. Weitere Informationen finden Sie unter Amazon-EC2-Schlüsselpaare im Amazon-EC2-Benutzerhandbuch für Linux-Instances.
   - SSHLocation: Gibt den IP-Adressbereich an, über den eine Verbindung mit der Instance über SSH hergestellt werden kann. Der Standard erlaubt den Zugriff von jeder IP-Adresse aus.

6. Auf der Seite Options (Optionen) können Sie auswählen, Ihre Stack-Ressourcen zu markieren. Wählen Sie Weiter aus.
7. Überprüfen Sie auf der Seite Review (Überprüfen) Ihre Informationen, bestätigen Sie, dass der Stack IAM-Ressourcen erstellen kann, und wählen Sie dann Create (Erstellen) aus.

   Wenn Sie die Konsole aktualisieren, sehen Sie, dass der neue Stack den `CREATE_IN_PROGRESS` Status hat.
8. Wenn die Instance erstellt wird, können Sie sie in der Amazon-EC2-Konsole sehen. Optional können Sie sich mit dem Host verbinden und den Fortschritt überprüfen.

   Verwenden Sie den folgenden Befehl, um zu bestätigen, dass der Agent installiert ist:

   ```
   rpm -qa amazon-cloudwatch-agent
   ```

   Verwenden Sie den folgenden Befehl, um zu bestätigen, dass der Agent ausgeführt wird:

   ```
   ps aux | grep amazon-cloudwatch-agent
   ```

Das nächste Verfahren zeigt, wie Sie den CloudWatch Agenten mithilfe AWS CloudFormation einer Inline-Vorlage aktualisieren können. Die standardmäßige Inline-Vorlage erfasst die `mem_used_percent`-Metrik. In diesem Tutorial ändern Sie die Agent-Konfiguration, um die Erfassung dieser Metrik zu stoppen.

Wird verwendet AWS CloudFormation , um den CloudWatch Agenten zu aktualisieren

1. Entfernen Sie in der Vorlage, die Sie im vorherigen Verfahren heruntergeladen haben, die folgenden Zeilen und speichern Sie die Vorlage:

   ```
   "mem": {

        "measurement": [
            "mem_used_percent"
          ]
    },
   ```
2. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.

3. Wählen Sie im AWS CloudFormation Dashboard den Stack aus, den Sie erstellt haben, und wählen Sie Stack aktualisieren.
4. Wählen Sie für Select Template (Vorlage auswählen) Upload a template to Amazon S3 (Vorlage auf Amazon S3 hochladen), wählen Sie die von Ihnen modifizierte Vorlage aus und klicken Sie auf Next (Weiter).
5. Wählen Sie auf der Seite Options (Optionen) die Option Next (Weiter) gefolgt von Next (Weiter) aus.
6. Prüfen Sie auf der Seite Review (Überprüfen) die Daten, und wählen Sie Update (Aktualisieren) aus.

   Nach einiger Zeit sehen Sie `UPDATE_COMPLETE`.

Tutorial: Installieren Sie den CloudWatch Agenten mithilfe von AWS CloudFormation Parameter Store

In diesem Tutorial erfahren Sie AWS CloudFormation , wie Sie den CloudWatch Agenten auf einer neuen Amazon EC2 EC2-Instance installieren. Dieses Tutorial wird auf einer neuen Instance installiert, die Amazon Linux 2 mit einer Agent-Konfigurationsdatei ausführt, die Sie in Parameter Store erstellt und gespeichert haben.

Nach der Vorgehensweise zur Installation des Agenten fährt das Tutorial mit der Aktualisierung des Agenten fort.

Wird verwendet AWS CloudFormation , um den CloudWatch Agenten mithilfe einer Konfiguration aus dem Parameter Store auf einer neuen Instance zu installieren

1. Falls Sie dies noch nicht getan haben, laden Sie das CloudWatch Agentenpaket auf einen Ihrer Computer herunter, damit Sie die Agenten-Konfigurationsdatei erstellen können. Weitere Informationen zum Herunterladen des Agenten mittels Parameter Store finden Sie unter Laden Sie den CloudWatch Agenten herunter und konfigurieren Sie ihn. Weitere Informationen zum Herunterladen des Pakets mithilfe der Befehlszeile finden Sie unter Laden Sie den CloudWatch Agenten über die Befehlszeile herunter und konfigurieren Sie ihn.
2. Erstellen Sie die Agentenkonfigurationsdatei und speichern Sie sie in Parameter Store. Weitere Informationen finden Sie unter Erstellen Sie die CloudWatch Agent-Konfigurationsdatei.
3. Laden Sie die Vorlage GitHub wie folgt herunter:

```
curl -O https://raw.githubusercontent.com/awslabs/aws-cloudformation-templates/
master/aws/solutions/AmazonCloudWatchAgent/ssm/amazon_linux.template
```

4. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
5. Wählen Sie Stack erstellen aus.
6. Wählen Sie für Choose a template (Vorlage auswählen) Upload a template to Amazon S3 (Vorlage auf Amazon S3 hochladen), wählen Sie die Vorlage aus, die Sie heruntergeladen haben, und klicken Sie auf Next (Weiter).
7. Füllen Sie auf der Seite Specify Details (Details angeben) die folgenden Parameter entsprechend aus, und klicken Sie dann auf Next (Weiter):

   - Stack-Name: Wählen Sie einen Stack-Namen für Ihren AWS CloudFormation Stack.
   - iamRole: Wählen Sie eine IAM-Rolle aus, die berechtigt ist, CloudWatch Metriken, Logs und Traces zu schreiben. Weitere Informationen finden Sie unter Erstellen Sie IAM-Rollen zur Verwendung mit dem CloudWatch Agenten auf Amazon EC2 EC2-Instances.
   - InstanceAMI: Wählen Sie ein AMI, das in der Region gültig ist, in der Sie Ihren Stack starten werden.
   - InstanceType: Wählen Sie einen gültigen Instance-Typ.
   - KeyName: Um den SSH-Zugriff auf die neue Instance zu aktivieren, wählen Sie ein vorhandenes Amazon EC2 EC2-Schlüsselpaar aus. Wenn Sie noch kein Amazon-EC2-Schlüsselpaar haben, können Sie eines in der AWS Management Console erstellen. Weitere Informationen finden Sie unter Amazon-EC2-Schlüsselpaare im Amazon-EC2-Benutzerhandbuch für Linux-Instances.
   - SSHLocation: Gibt den IP-Adressbereich an, über den eine Verbindung mit der Instance über SSH hergestellt werden kann. Der Standard erlaubt den Zugriff von jeder IP-Adresse aus.
   - SSMKey: Gibt die Agent-Konfigurationsdatei an, die Sie in Parameter Store erstellt und gespeichert haben.
8. Auf der Seite Options (Optionen) können Sie auswählen, Ihre Stack-Ressourcen zu markieren. Wählen Sie Weiter aus.
9. Überprüfen Sie auf der Seite Review (Überprüfen) Ihre Informationen, bestätigen Sie, dass der Stack IAM-Ressourcen erstellen kann, und wählen Sie dann Create (Erstellen) aus.

   Wenn Sie die Konsole aktualisieren, sehen Sie, dass der neue Stack den `CREATE_IN_PROGRESS` Status hat.

10. Wenn die Instance erstellt wird, können Sie sie in der Amazon-EC2-Konsole sehen. Optional können Sie sich mit dem Host verbinden und den Fortschritt überprüfen.

    Verwenden Sie den folgenden Befehl, um zu bestätigen, dass der Agent installiert ist:

    ```
    rpm -qa amazon-cloudwatch-agent
    ```

    Verwenden Sie den folgenden Befehl, um zu bestätigen, dass der Agent ausgeführt wird:

    ```
    ps aux | grep amazon-cloudwatch-agent
    ```

Das nächste Verfahren zeigt, wie Sie den CloudWatch Agenten mithilfe AWS CloudFormation einer Agentenkonfiguration aktualisieren, die Sie im Parameter Store gespeichert haben.

Wird verwendet AWS CloudFormation , um den CloudWatch Agenten mithilfe einer Konfiguration im Parameter Store zu aktualisieren

1. Ändern Sie die Agentenkonfigurationsdatei, die in Parameter Store gespeichert ist, auf die neue Konfiguration, die Sie wünschen.
2. Ändern Sie in der AWS CloudFormation Vorlage, die Sie im the section called “Tutorial: Installieren Sie den CloudWatch Agenten mithilfe von AWS CloudFormation Parameter Store” Thema heruntergeladen haben, die Versionsnummer. Sie können z. B. `VERSION=1.0` zu `VERSION=2.0` ändern.
3. Öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
4. Wählen Sie im AWS CloudFormation Dashboard den Stack aus, den Sie erstellt haben, und wählen Sie Stack aktualisieren.
5. Wählen Sie für Select Template (Vorlage auswählen) Upload a template to Amazon S3 (Vorlage auf Amazon S3 hochladen), wählen Sie die Vorlage aus, die Sie gerade geändert haben, und klicken Sie auf Next (Weiter).
6. Wählen Sie auf der Seite Options (Optionen) die Option Next (Weiter) gefolgt von Next (Weiter) aus.
7. Prüfen Sie auf der Seite Review (Überprüfen) die Daten, und wählen Sie Update (Aktualisieren) aus.

   Nach einiger Zeit sehen Sie `UPDATE_COMPLETE`.

Fehlerbehebung bei der Installation des CloudWatch Agenten mit AWS CloudFormation

Dieser Abschnitt hilft Ihnen bei der Behebung von Problemen bei der Installation und Aktualisierung des CloudWatch Agenten mithilfe von AWS CloudFormation.

Erkennen, wenn eine Aktualisierung fehlschlägt

Wenn Sie Ihre CloudWatch Agentenkonfiguration aktualisieren und eine ungültige Konfiguration verwenden, sendet der Agent keine Metriken mehr an CloudWatch. AWS CloudFormation Eine schnelle Möglichkeit, um zu überprüfen, ob ein Update der Agentenkonfiguration erfolgreich war, ist ein Blick in die Datei `cfn-init-cmd.log`. Auf einem Linux-Server befindet sich die Datei unter `/var/log/cfn-init-cmd.log`. Auf einer Windows-Instance befindet sich die Datei unter `C:\cfn\log\cfn-init-cmd.log`.

Metriken fehlen

Wenn Sie die nach der Installation oder Aktualisierung des Agenten erwarteten Metriken nicht sehen, stellen Sie sicher, dass der Agent so konfiguriert ist, dass er diese Metrik erfasst. Überprüfen Sie dazu die Datei `amazon-cloudwatch-agent.json`, um sicherzustellen, dass die Metrik aufgelistet ist. Prüfen Sie auch, ob die Suche im korrekten Metrik-Namespace erfolgt. Weitere Informationen finden Sie unter CloudWatch Agentendateien und Speicherorte.

CloudWatch Präferenz für Agenten-Anmeldeinformationen

In diesem Abschnitt wird die Anbieterkette für Anmeldeinformationen beschrieben, die der CloudWatch Agent verwendet, um Anmeldeinformationen bei der Kommunikation mit anderen AWS Diensten und APIs abzurufen. Die Reihenfolge ist wie folgt. Die in den Nummern zwei bis fünf der folgenden Liste aufgeführten Einstellungen entsprechen der im AWS SDK definierten Reihenfolge. Weitere Informationen finden Sie in der SDK-Dokumentation unter Anmeldeinformationen angeben.

1. Gemeinsam genutzte Konfigurations- und Anmeldeinformationsdateien, wie sie in der CloudWatch `common-config.toml` Agentendatei definiert sind. Weitere Informationen finden Sie unter (Optional) Ändern der gemeinsamen Konfiguration für Proxy- oder Regionsangaben.
2. AWS SDK-Umgebungsvariablen

> **Important**
>
> Wenn Sie unter Linux den CloudWatch Agenten mithilfe des `amazon-cloudwatch-agent-ctl` Skripts ausführen, startet das Skript den Agenten als `systemd`

Dienst. In diesem Fall kann der Agent nicht auf Umgebungsvariablen wie `HOMEAWS_ACCESS_KEY_ID`, und `AWS_SECRET_ACCESS_KEY` zugreifen.

3. Gemeinsam genutzte Konfigurations- und Anmeldeinformationsdateien befinden sich in `$HOME/%USERPROFILE%`

> **Note**
>
> Der CloudWatch Agent sucht nach `.aws/credentials` in `$HOME` für Linux und macOS und sucht `%USERPROFILE%` nach Windows. Im Gegensatz zum AWS SDK verfügt der CloudWatch Agent nicht über Fallback-Methoden, um das Home-Verzeichnis zu ermitteln, falls auf die Umgebungsvariablen nicht zugegriffen werden kann. Dieser Unterschied im Verhalten besteht darin, die Abwärtskompatibilität mit früheren Implementierungen des SDK aufrechtzuerhalten. AWS
> Wenn die vom AWS SDK abgeleiteten gemeinsamen Anmeldeinformationen ablaufen und rotiert werden`common-config.toml`, werden die erneuerten Anmeldeinformationen im Gegensatz zu den gemeinsamen Anmeldeinformationen in außerdem nicht automatisch vom CloudWatch Agenten übernommen und erfordern dazu einen Neustart des Agenten.

4. Eine AWS Identity and Access Management Rolle für Aufgaben, wenn eine Anwendung vorhanden ist, die eine Amazon Elastic Container Service-Aufgabendefinition oder eine RunTask API-Operation verwendet.
5. Ein einer Amazon EC2-Instance hinzugefügtes Instance-Profil.

Als bewährte Methode empfehlen wir, dass Sie die Anmeldeinformationen in der folgenden Reihenfolge angeben, wenn Sie den CloudWatch Agenten verwenden.

1. Verwenden Sie IAM-Rollen für Aufgaben, wenn Ihre Anwendung eine Amazon Elastic Container Service-Aufgabendefinition oder eine RunTask API-Operation verwendet.
2. Verwenden Sie IAM-Rollen, wenn Ihre Anwendung auf einer Amazon EC2 EC2-Instance ausgeführt wird.
3. Verwenden Sie die CloudWatch `common-config.toml` Agentendatei, um die Anmeldeinformationsdatei anzugeben. Diese Anmeldeinformationsdatei ist dieselbe, die von anderen AWS SDKs und dem AWS CLI verwendet wird. Wenn Sie bereits eine Datei mit gemeinsam genutzten Anmeldeinformationen verwenden, können Sie sie auch für diesen Zweck verwenden. Wenn Sie sie mithilfe der CloudWatch `common-config.toml` Agentendatei

bereitstellen, stellen Sie sicher, dass der Agent die Anmeldeinformationen nach deren Ablauf rotiert und ersetzt, ohne dass Sie den Agenten neu starten müssen.

4. Umgebungsvariablen verwenden. Das Setzen von Umgebungsvariablen ist nützlich, wenn Sie Entwicklungsarbeiten auf einem anderen Computer als einer Amazon EC2 EC2-Instance durchführen.

> **Note**
>
> Wenn Sie Telemetriedaten an ein anderes Konto senden, wie unter beschriebenMetriken, Protokolle und Ablaufverfolgungen an ein anderes Konto senden, verwendet der CloudWatch Agent die in diesem Abschnitt beschriebene Anbieterkette für Anmeldeinformationen, um die ersten Anmeldeinformationen abzurufen. Anschließend verwendet er diese Anmeldeinformationen, wenn er die `role_arn` in der CloudWatch Agentenkonfigurationsdatei angegebene IAM-Rolle annimmt.

Überprüfung der Signatur des Agentenpakets CloudWatch

GPG-Signaturdateien sind für CloudWatch Agentenpakete auf Linux-Servern enthalten. Sie können einen öffentlichen Schlüssel verwenden, um sicherzustellen, dass die Download-Datei des Agenten original und unverändert ist.

Für Windows Server können Sie den MSI verwenden, um die Signatur zu überprüfen.

Bei macOS-Computern ist die Signatur im Agenten-Download-Paket enthalten.

Die richtige Signaturdatei finden Sie in der folgenden Tabelle. Für jede Architektur und jedes Betriebssystem gibt es einen allgemeinen Link sowie Links für jede Region. Für Amazon Linux 2023 und Amazon Linux 2 und die x86-64-Architektur lauten beispielsweise drei der gültigen Links:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig`
- `https://amazoncloudwatch-agent-us-east-1.s3.us-east-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-eu-central-1.s3.eu-central-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

> **Note**
>
> Um den CloudWatch Agenten herunterzuladen, muss Ihre Verbindung TLS 1.2 oder höher verwenden.

Architektur	Plattform	Download-Link	Link zur Signaturdatei
x86-64	Amazon Linux 2023 und Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/amazon_linux/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/amazon_linux/amd64/latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	Centos	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/centos/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/centos/amd64/latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/	https://amazoncloudwatch-agent.s3.amazonaws.com/

Architektur	Plattform	Download-Link	Link zur Signaturdatei
		redhat/amd64/latest/ amazon-cloudwatch-agent .rpm	redhat/amd64/latest/ amazon-cloudwatch-agent .rpm.sig
		https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/redhat/amd64/latest/* .rpm amazon-cloudwatch-agent	*https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/redhat/amd64/latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/ amazon-cloudwatch-agent .rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/ amazon-cloudwatch-agent .rpm.sig
		https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/suse/amd64/latest/* .rpm amazon-cloudwatch-agent	*https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/suse/amd64/latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	Debian	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/ amazon-cloudwatch-agent .deb	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/ amazon-cloudwatch-agent .deb.sig
		https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/debian/amd64/latest/* .deb amazon-cloudwatch-agent	*https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/debian/amd64/latest/* .deb.sig amazon-cloudwatch-agent

Architektur	Plattform	Download-Link	Link zur Signaturdatei
x86-64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/ubuntu/amd64/latest/* .deb amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/ amazon-cloudwatch-agent .deb.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/ubuntu/amd64/latest/* .deb.sig amazon-cloudwatch-agent
x86-64	Oracle	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/oracle_linux/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/oracle_linux/amd64/latest/* .rpm.sig amazon-cloudwatch-agent

Architektur	Plattform	Download-Link	Link zur Signaturdatei
x86-64	macOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/ amazon-cloudwatch-agent .pkg https://amazoncloudwatch-agent - *Region .s3. region .amazonaws.com/darwin/amd64 /latest/* .pkg amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/ amazon-cloudwatch-agent .pkg.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/darwin/amd64/latest/* .pkg.sig amazon-cloudwatch-agent
x86-64	Windows	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/ amazon-cloudwatch-agent .msi https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/windows/amd64/latest/* .msi amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/ amazon-cloudwatch-agent .msi.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/windows/amd64/latest/* .msi.sig amazon-cloudwatch-agent

Architektur	Plattform	Download-Link	Link zur Signaturdatei
ARM64	Amazon Linux 2023 und Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3.region .amazonaws.com/amazon_linux/arm64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3.region .amazonaws.com/amazon_linux/arm64/latest/* .rpm.sig amazon-cloudwatch-agent
ARM64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3.region .amazonaws.com/redhat/arm64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3.region .amazonaws.com/redhat/arm64/latest/* .rpm.sig amazon-cloudwatch-agent

Architektur	Plattform	Download-Link	Link zur Signaturdatei
ARM64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/ubuntu/arm64/latest/* .deb amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/ amazon-cloudwatch-agent .deb.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/ubuntu/arm64/latest/* .deb.sig amazon-cloudwatch-agent
ARM64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent — *Region .s3. region .amazonaws.com/suse/arm64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent — Region .s3. region .amazonaws.com/suse/arm64/latest/* .rpm.sig amazon-cloudwatch-agent

Um das Agentenpaket auf einem Linux-Server zu verifizieren CloudWatch

1. Laden Sie den öffentlichen Schlüssel herunter.

```
shell$ wget https://amazoncloudwatch-agent.s3.amazonaws.com/assets/amazon-cloudwatch-agent.gpg
```

2. Importieren Sie den öffentlichen Schlüssel in Ihren Schlüsselbund.

```
shell$  gpg --import amazon-cloudwatch-agent.gpg
```

```
gpg: key 3B789C72: public key "Amazon CloudWatch Agent" imported
gpg: Total number processed: 1
gpg: imported: 1 (RSA: 1)
```

Notieren Sie sich den Schlüsselwert. Sie brauchen ihn im nächsten Schritt. Im vorangegangenen Beispiel ist der Schlüsselwert `3B789C72`.

3. Überprüfen Sie den Fingerabdruck, indem Sie den folgenden Befehl ausführen und *Schlüssel-Wert* durch den Wert des vorherigen Schritts ersetzen:

```
shell$  gpg --fingerprint key-value
pub   2048R/3B789C72 2017-11-14
      Key fingerprint = 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
uid                  Amazon CloudWatch Agent
```

Die Fingerabdruck-Zeichenfolge muss der folgenden entsprechen:

```
9376 16F3 450B 7D80 6CBD 9725 D581 6730 3B78 9C72
```

Wenn die Fingerabdruck-Zeichenfolge nicht übereinstimmt, installieren Sie den Agent nicht. Wenden Sie sich an Amazon Web Services.

Nachdem Sie den Fingerabdruck verifiziert haben, können Sie ihn verwenden, um die Signatur des CloudWatch Agentenpakets zu überprüfen.

4. Laden Sie die Paket-Signaturdatei mittels wget herunter. Um die richtige Signaturdatei zu bestimmen, lesen Sie die vorherige Tabelle.

```
wget Signature File Link
```

5. Führen Sie gpg --verify aus, um die Signatur zu überprüfen.

```
shell$ gpg --verify signature-filename agent-download-filename
gpg: Signature made Wed 29 Nov 2017 03:00:59 PM PST using RSA key ID 3B789C72
gpg: Good signature from "Amazon CloudWatch Agent"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
```

Wenn die Ausgabe die Bezeichnung `BAD signature`enthält, überprüfen Sie, ob Sie das Verfahren korrekt durchgeführt haben. Wenn Sie diese Antwort weiterhin erhalten, wenden Sie

sich bitte an Amazon Web Services und vermeiden Sie die Verwendung der heruntergeladenen Datei.

Beachten Sie die Warnung zu vertrauenswürdigen Inhalten. Beachten Sie die Warnung zu vertrauenswürdigen Inhalten. Das bedeutet nicht, dass die Signatur ungültig ist, sondern nur, dass Sie den öffentlichen Schlüssel nicht überprüft haben.

Um das CloudWatch Agentenpaket auf einem Server zu überprüfen, auf dem Windows Server ausgeführt wird

1. Laden Sie GnuPG für Windows unter https://gnupg.org/download/ herunter und installieren Sie es. Fügen Sie bei der Installation die Option Shell Extension (GpgEx) hinzu.

   Sie können die verbleibenden Schritte in Windows ausführen PowerShell.

2. Laden Sie den öffentlichen Schlüssel herunter.

   ```
   PS> wget https://amazoncloudwatch-agent.s3.amazonaws.com/assets/amazon-cloudwatch-agent.gpg -OutFile amazon-cloudwatch-agent.gpg
   ```

3. Importieren Sie den öffentlichen Schlüssel in Ihren Schlüsselbund.

   ```
   PS>  gpg --import amazon-cloudwatch-agent.gpg
   gpg: key 3B789C72: public key "Amazon CloudWatch Agent" imported
   gpg: Total number processed: 1
   gpg: imported: 1 (RSA: 1)
   ```

   Notieren Sie sich den Schlüsselwert. Sie benötigen ihn im nächsten Schritt. Im vorangegangenen Beispiel ist der Schlüsselwert `3B789C72`.

4. Überprüfen Sie den Fingerabdruck, indem Sie den folgenden Befehl ausführen und *`Schlüssel-Wert`* durch den Wert des vorherigen Schritts ersetzen:

   ```
   PS>  gpg --fingerprint key-value
   pub   rsa2048 2017-11-14 [SC]
         9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
   uid           [ unknown] Amazon CloudWatch Agent
   ```

   Die Fingerabdruck-Zeichenfolge muss der folgenden entsprechen:

   ```
   9376 16F3 450B 7D80 6CBD 9725 D581 6730 3B78 9C72
   ```

Wenn die Fingerabdruck-Zeichenfolge nicht übereinstimmt, installieren Sie den Agent nicht. Wenden Sie sich an Amazon Web Services.

Nachdem Sie den Fingerabdruck verifiziert haben, können Sie ihn verwenden, um die Signatur des CloudWatch Agentenpakets zu überprüfen.

5. Laden Sie die Paket-Signaturdatei mit wget herunter. Informationen zum Ermitteln der richtigen Signaturdatei finden Sie unter Download-Links für CloudWatch Agenten.
6. Führen Sie gpg --verify aus, um die Signatur zu überprüfen.

```
PS> gpg --verify sig-filename agent-download-filename
gpg: Signature made 11/29/17 23:00:45 Coordinated Universal Time
gpg:                using RSA key D58167303B789C72
gpg: Good signature from "Amazon CloudWatch Agent" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
```

Wenn die Ausgabe die Bezeichnung `BAD signature`enthält, überprüfen Sie, ob Sie das Verfahren korrekt durchgeführt haben. Wenn Sie diese Antwort weiterhin erhalten, wenden Sie sich bitte an Amazon Web Services und vermeiden Sie die Verwendung der heruntergeladenen Datei.

Beachten Sie die Warnung zu vertrauenswürdigen Inhalten. Beachten Sie die Warnung zu vertrauenswürdigen Inhalten. Das bedeutet nicht, dass die Signatur ungültig ist, sondern nur, dass Sie den öffentlichen Schlüssel nicht überprüft haben.

So verifizieren Sie das CloudWatch Agentenpaket auf einem macOS-Computer

- Es gibt zwei Methoden zur Signaturenverifizierung unter macOS.
  - Überprüfen Sie den Fingerabdruck, indem Sie den folgenden Befehl ausführen.

```
pkgutil --check-signature amazon-cloudwatch-agent.pkg
```

Das Ergebnis sollte in etwa wie folgt aussehen.

```
Package "amazon-cloudwatch-agent.pkg":
```

```
       Status: signed by a developer certificate issued by Apple for
distribution
       Signed with a trusted timestamp on: 2020-10-02 18:13:24 +0000
       Certificate Chain:
       1. Developer ID Installer: AMZN Mobile LLC (94KV3E626L)
       Expires: 2024-10-18 22:31:30 +0000
       SHA256 Fingerprint:
       81 B4 6F AF 1C CA E1 E8 3C 6F FB 9E 52 5E 84 02 6E 7F 17 21 8E FB
       0C 40 79 13 66 8D 9F 1F 10 1C

------------------------------------------------------------------------
       2. Developer ID Certification Authority
       Expires: 2027-02-01 22:12:15 +0000
       SHA256 Fingerprint:
       7A FC 9D 01 A6 2F 03 A2 DE 96 37 93 6D 4A FE 68 09 0D 2D E1 8D 03
       F2 9C 88 CF B0 B1 BA 63 58 7F

------------------------------------------------------------------------
       3. Apple Root CA
       Expires: 2035-02-09 21:40:36 +0000
       SHA256 Fingerprint:
       B0 B1 73 0E CB C7 FF 45 05 14 2C 49 F1 29 5E 6E DA 6B CA ED 7E 2C
       68 C5 BE 91 B5 A1 10 01 F0 24
```

- Oder laden Sie die SIG-Datei herunter und verwenden Sie sie. Gehen Sie folgendermaßen vor, um diese Methode zu verwenden.
  - Installieren Sie die GPG-Anwendung auf Ihrem macOS-Host, indem Sie den folgenden Befehl eingeben.

    ```
    brew install GnuPG
    ```

- Laden Sie die Paket-Signaturdatei mittels Curl herunter. Informationen zum Ermitteln der richtigen Signaturdatei finden Sie unter Download-Links für CloudWatch Agenten.
- Führen Sie gpg --verify aus, um die Signatur zu überprüfen.

```
PS> gpg --verify sig-filename agent-download-filename
gpg: Signature made 11/29/17 23:00:45 Coordinated Universal Time
gpg:                using RSA key D58167303B789C72
gpg: Good signature from "Amazon CloudWatch Agent" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
```

```
Primary key fingerprint: 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
```

Wenn die Ausgabe die Bezeichnung `BAD signature`enthält, überprüfen Sie, ob Sie das Verfahren korrekt durchgeführt haben. Wenn Sie diese Antwort weiterhin erhalten, wenden Sie sich bitte an Amazon Web Services und vermeiden Sie die Verwendung der heruntergeladenen Datei.

Beachten Sie die Warnung zu vertrauenswürdigen Inhalten. Beachten Sie die Warnung zu vertrauenswürdigen Inhalten. Das bedeutet nicht, dass die Signatur ungültig ist, sondern nur, dass Sie den öffentlichen Schlüssel nicht überprüft haben.

Erstellen Sie die CloudWatch Agent-Konfigurationsdatei

Bevor Sie den CloudWatch Agenten auf einem beliebigen Server ausführen können, müssen Sie eine oder mehrere CloudWatch Agentenkonfigurationsdateien erstellen.

Die Konfigurationsdatei des Agenten ist eine JSON-Datei, in der die Metriken, Protokolle und Ablaufverfolgungen angegeben sind, die der Agent erfassen soll, einschließlich benutzerdefinierter Metriken. Sie können sie mithilfe des Assistenten oder selbst von Grund auf erstellen. Sie können die Konfigurationsdatei auch mit dem Assistenten erstellen und dann manuell anpassen. Wenn Sie die Datei manuell erstellen oder bearbeiten, ist der Prozess komplizierter. Sie haben jedoch mehr Kontrolle über die erfassten Metriken und können Metriken angeben, die im Assistenten nicht verfügbar sind.

Bei jeder Änderung der Agent-Konfigurationsdatei müssen Sie den Agent neu starten, damit die Änderungen wirksam werden. Um den Agent neu zu starten, befolgen Sie die Anweisungen in Starten Sie den Agenten CloudWatch .

Sie können die erstellte Konfigurationsdatei als JSON-Datei speichern und später für die Installation des Agenten auf Ihren Servern verwenden. Alternativ können Sie die Datei in Systems Manager Parameter Store speichern, wenn Sie für die Agenteninstallation auf den Servern Systems Manager verwenden möchten.

Der CloudWatch Agent unterstützt die Verwendung mehrerer Konfigurationsdateien. Weitere Informationen finden Sie unter Mehrere CloudWatch Agenten-Konfigurationsdateien.

Für die vom CloudWatch Agenten gesammelten Metriken, Protokolle und Traces fallen Gebühren an. Weitere Informationen zur Preisgestaltung finden Sie unter CloudWatch Amazon-Preise.

Inhalt

Erstellen Sie die CloudWatch Agenten-Konfigurationsdatei mit dem Assistenten

Der Assistent für die Agentenkonfigurationsdatei`amazon-cloudwatch-agent-config-wizard`,, stellt eine Reihe von Fragen, um Ihnen bei der Konfiguration des CloudWatch Agenten für Ihre Bedürfnisse zu helfen.

Erforderliche Anmeldeinformationen

Der Assistent kann die zu verwendenden Anmeldeinformationen und die zu verwendende AWS Region automatisch erkennen, wenn Sie die AWS Anmeldeinformationen und Konfigurationsdateien vor dem Start des Assistenten eingerichtet haben. Weitere Informationen zu diesen Dateien finden Sie unter Konfigurations- und Anmeldeinformationsdateien im AWS Systems Manager - Benutzerhandbuch.

In der AWS Anmeldeinformationsdatei sucht der Assistent nach Standardanmeldedaten und sucht auch nach einem `AmazonCloudWatchAgent` Abschnitt wie dem folgenden:

```
[AmazonCloudWatchAgent]
aws_access_key_id = my_access_key
aws_secret_access_key = my_secret_key
```

Der Assistent zeigt die Standard-Anmeldeinformationen, die Anmeldeinformationen aus `AmazonCloudWatchAgent` und die Option `Others` an. Sie können auswählen, welche Anmeldeinformationen verwendet werden sollen. Bei Wahl von `Others` (Andere), können Sie Anmeldeinformationen eingeben.

Verwenden Sie anstelle von *`my_access_key`* und *`my_secret_key`* die Schlüssel des IAM-Benutzers, der Schreibrechte für Systems Manager Parameter Store hat. Weitere Informationen zu den IAM-Benutzern, die für den CloudWatch Agenten benötigt werden, finden Sie unterErstellen Sie IAM-Benutzer zur Verwendung mit dem CloudWatch Agenten auf lokalen Servern.

In der AWS Konfigurationsdatei können Sie die Region angeben, an die der Agent Metriken sendet, falls es sich um eine andere Region als den `[default]` Abschnitt handelt. Die Voreinstellung ist,

die Metriken in der Region zu veröffentlichen, in der sich die Amazon-EC2-Instance befindet. Wenn die Metriken in einer anderen Region veröffentlicht werden sollen, geben Sie hier die Region an. Im folgenden Beispiel werden die Metriken in der Region `us-west-1` veröffentlicht.

```
[AmazonCloudWatchAgent]
region = us-west-1
```

Führen Sie den Assistenten zur CloudWatch Agentenkonfiguration aus

Um die CloudWatch Agenten-Konfigurationsdatei zu erstellen

1. Starten Sie den Assistenten zur CloudWatch Agentenkonfiguration, indem Sie Folgendes eingeben:

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard
   ```

   Führen Sie auf einem Server mit Windows Server die folgenden Befehle aus, um den Assistenten zu starten:

   ```
   cd "C:\Program Files\Amazon\AmazonCloudWatchAgent"
   ```

   ```
   .\amazon-cloudwatch-agent-config-wizard.exe
   ```

2. Beantworten Sie die Fragen zum Anpassen der Konfigurationsdatei für Ihren Server.
3. Wenn Sie die Konfigurationsdatei lokal speichern, wird die Konfigurationsdatei `config.json` in `/opt/aws/amazon-cloudwatch-agent/bin/` auf Linux-Servern und in `C:\Program Files\Amazon\AmazonCloudWatchAgent` auf Windows Server-Servern gespeichert. Anschließend können Sie diese Datei auf andere Server kopieren, auf denen der Agent installiert werden soll.

   Wenn Sie Systems Manager zum Installieren und Konfigurieren des Agenten verwenden, müssen Sie mit Yes (Ja) antworten, wenn Sie gefragt werden, ob Sie die Datei in Systems Manager Parameter Store speichern möchten. Sie können sich auch dafür entscheiden, die Datei im Parameter Store zu speichern, auch wenn Sie den SSM-Agent nicht zur Installation des CloudWatch Agenten verwenden. Zum Speichern der Datei in Parameter Store müssen Sie eine IAM-Rolle mit ausreichenden Berechtigungen verwenden. Weitere Informationen finden Sie unter Erstellen Sie IAM-Rollen und -Benutzer für die Verwendung mit dem Agenten CloudWatch .

CloudWatch Vordefinierte Metriksätze für Agenten

Der Assistent ist mit vordefinierten Metrikkategorien mit unterschiedlichen Detailebenen konfiguriert. Diese Metrikkategorien werden in den folgenden Tabellen dargestellt. Weitere Informationen zu diesen Metriken finden Sie unter Vom CloudWatch Agenten gesammelte Metriken.

> **Note**
>
> Parameter Store unterstützt Parameter in den Stufen „Standard“ und „Advanced“. Diese Parameterebenen beziehen sich nicht auf die Ebenen „Basic“, „Standard“ und „Advanced“ der Metrikdetails, die in diesen Tabellen beschrieben werden.

Amazon-EC2-Instances mit Linux

Detailstufe	Enthaltene Metriken
Basic	Mem: mem_used_percent Disk: disk_used_percent Die `disk`-Metriken wie `disk_used_percent` haben eine Dimension für `Partition`, was bedeutet, dass die Anzahl der generierten benutzerdefinierten Metriken von der Anzahl der Partitionen abhängt, die Ihrer Instance zugeordnet sind. Die Anzahl der Festplattenpartitionen hängt davon ab, welches AMI Sie verwenden, und wie viele Amazon-EBS-Volumes Sie an den Server anfügen.
Standard	CPU: `cpu_usage_idle`, `cpu_usage_iowait`, `cpu_usage_user`, `cpu_usage_system` Disk: `disk_used_percent`, `disk_inodes_free` Diskio: `diskio_io_time` Mem: `mem_used_percent` Swap: `swap_used_percent`

Detailstufe	Enthaltene Metriken
Advanced	CPU: `cpu_usage_idle` , `cpu_usage_iowait` , `cpu_usage_user` , `cpu_usage_system` Disk: `disk_used_percent` , `disk_inodes_free` Diskio: `diskio_io_time` , `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads` Mem: `mem_used_percent` Netstat: `netstat_tcp_established` , `netstat_tcp_time_wait` Swap: `swap_used_percent`

On-Premises-Server mit Linux

Detailstufe	Enthaltene Metriken
Basic	Disk: `disk_used_percent` Diskio: `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads` Mem: `mem_used_percent` Net: `net_bytes_sent` , `net_bytes_recv` , `net_packets_sent` , `net_packets_recv` Swap: `swap_used_percent`
Standard	CPU: `cpu_usage_idle` , `cpu_usage_iowait` Disk: `disk_used_percent` , `disk_inodes_free` Diskio: `diskio_io_time` , `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads` Mem: `mem_used_percent`

Detailstufe	Enthaltene Metriken
	Net: `net_bytes_sent` , `net_bytes_recv` , `net_packets_sent` , `net_packets_recv` Swap: `swap_used_percent`
Advanced	CPU: `cpu_usage_guest` , `cpu_usage_idle` , `cpu_usage_iowait` , `cpu_usage_steal` , `cpu_usage_user` , `cpu_usage_system` Disk: `disk_used_percent` , `disk_inodes_free` Diskio: `diskio_io_time` , `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads` Mem: `mem_used_percent` Net: `net_bytes_sent` , `net_bytes_recv` , `net_packets_sent` , `net_packets_recv` Netstat: `netstat_tcp_established` , `netstat_tcp_time_wait` Swap: `swap_used_percent`

Amazon-EC2-Instances mit Windows Server

> **Note**
>
> Die in dieser Tabelle aufgeführten Metriknamen zeigen an, wie die Metrik in der Konsole angezeigt wird. Der tatsächliche Name der Metrik enthält möglicherweise nicht das erste Wort. Der tatsächliche Metrikname für `LogicalDisk % Free Space` lautet beispielsweise nur `% Free Space`.

Detailstufe	Enthaltene Metriken
Basic	Memory: `Memory % Committed Bytes In Use` LogicalDisk: `LogicalDisk % Free Space`

Detailstufe	Enthaltene Metriken
Standard	Memory: `Memory % Committed Bytes In Use` Paging: `Paging File % Usage` Processor: `Processor % Idle Time`, `Processor % Interrupt Time`, `Processor % User Time` PhysicalDisk: `PhysicalDisk % Disk Time` LogicalDisk: `LogicalDisk % Free Space`
Advanced	Memory: `Memory % Committed Bytes In Use` Paging: `Paging File % Usage` Processor: `Processor % Idle Time`, `Processor % Interrupt Time`, `Processor % User Time` LogicalDisk: `LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk % Disk Time` , `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` TCP: `TCPv4 Connections Established` , `TCPv6 Connections Established`

On-Premises-Server mit Windows Server

> **Note**
>
> Die in dieser Tabelle aufgeführten Metriknamen zeigen an, wie die Metrik in der Konsole angezeigt wird. Der tatsächliche Name der Metrik enthält möglicherweise nicht das erste Wort. Der tatsächliche Metrikname für `LogicalDisk % Free Space` lautet beispielsweise nur `% Free Space`.

Detailstufe	Enthaltene Metriken
Basic	Paging: `Paging File % Usage` Processor: `Processor % Processor Time` LogicalDisk:`LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` Memory: `Memory % Committed Bytes In Use` Network Interface: `Network Interface Bytes Sent/sec`, `Network Interface Bytes Received/sec` , `Network Interface Packets Sent/sec`, `Network Interface Packets Received/sec`
Standard	Paging: `Paging File % Usage` Processor: `Processor % Processor Time`, `Processor % Idle Time`, `Processor % Interrupt Time` LogicalDisk: `LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk % Disk Time` , `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` Memory: `Memory % Committed Bytes In Use` Network Interface: `Network Interface Bytes Sent/sec`, `Network Interface Bytes Received/sec` , `Network Interface Packets Sent/sec`, `Network Interface Packets Received/sec`
Advanced	Paging:`Paging File % Usage` Processor: `Processor % Processor Time`, `Processor % Idle Time`, `Processor % Interrupt Time`, `Processor % User Time`

Detailstufe	Enthaltene Metriken
	LogicalDisk: `LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk % Disk Time` , `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` Memory: `Memory % Committed Bytes In Use` Network Interface: `Network Interface Bytes Sent/sec`, `Network Interface Bytes Received/sec` , `Network Interface Packets Sent/sec`, `Network Interface Packets Received/sec` TCP: `TCPv4 Connections Established` , `TCPv6 Connections Established`

Erstellen oder bearbeiten Sie die CloudWatch Agenten-Konfigurationsdatei manuell

Die CloudWatch Agenten-Konfigurationsdatei ist eine JSON-Datei mit vier Abschnitten,`agent`,`metrics`, und `logstraces`, die wie folgt beschrieben werden:

- Der Abschnitt `agent` enthält Felder für die allgemeine Konfiguration des Agenten.
- `metrics`In diesem Abschnitt werden die benutzerdefinierten Messwerte für die Erfassung und Veröffentlichung angegeben CloudWatch. Wenn Sie den Agenten nur verwenden, um Protokolle zu erfassen, können Sie den Abschnitt `metrics` in der Datei weglassen.
- `logs`In diesem Abschnitt wird angegeben, welche Protokolldateien in CloudWatch Logs veröffentlicht werden. Hierbei kann es sich u. a. um Ereignisse aus dem Windows-Ereignisprotokoll handeln, wenn auf dem Server Windows Server ausgeführt wird.
- `traces`In diesem Abschnitt werden die Quellen für Traces angegeben, die gesammelt und an sie gesendet werden AWS X-Ray.

In den folgenden Abschnitten werden die Struktur und die Felder dieser JSON-Datei erläutert. Sie können auch die Schemadefinition für diese Konfigurationsdatei anzeigen. Die Schemadefinition

befindet sich auf Linux-Servern unter *installation-directory*/doc/amazon-cloudwatch-agent-schema.json und auf Servern mit Windows Server unter *installation-directory*/amazon-cloudwatch-agent-schema.json.

Wenn Sie die -Agentenkonfigurationsdatei manuell erstellen oder bearbeiten, können Sie ihr einen beliebigen Namen geben. Zur Vereinfachung der Fehlerbehebung wird empfohlen, ihr auf Linux-Servern den Namen /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json und auf Servern, auf denen Windows Server ausgeführt wird, den Namen $Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json zu geben. Anschließend können Sie die Datei auf andere Server kopieren, auf denen der Agent installiert werden soll.

> **Note**
>
> Für Metriken, Protokolle und Traces, die vom CloudWatch Agenten erfasst werden, fallen Gebühren an. Weitere Informationen zur Preisgestaltung finden Sie unter CloudWatch Amazon-Preise.

CloudWatch Agenten-Konfigurationsdatei: Abschnitt Agent

Der Abschnitt agent kann die folgenden Felder enthalten: Der Assistent erstellt keinen Abschnitt agent. Stattdessen lässt der Assistent diesen weg und verwendet die Standardwerte für alle Felder in diesem Abschnitt.

- metrics_collection_interval – Optional. Gibt an, wie oft alle in dieser Konfigurationsdatei angegebenen Metriken erfasst werden. Sie können den Wert für bestimmte Arten von Metriken überschreiben.

  Der Wert wird in Sekunden angegeben. Beispiel: Angeben, dass 10 Metriken alle 10 Sekunden und 300 Metriken alle 5 Minuten gesammelt werden sollen.

  Wenn Sie diesen Wert auf weniger als 60 Sekunden festlegen, wird die jeweilige Metrik als hochauflösende Metrik erfasst. Weitere Informationen zu hochauflösenden Metriken finden Sie unter Hochauflösende Metriken.

  Der Standardwert lautet 60.
- region— Gibt die Region an, die für den CloudWatch Endpunkt verwendet werden soll, wenn eine Amazon EC2 EC2-Instance überwacht wird. Die gesammelten Metriken werden an diese Region

gesendet, wie z. B. `us-west-1`. Wenn Sie dieses Feld weglassen, sendet der Agent Metriken an die Region, in der sich die Amazon-EC2-Instance befindet.

Wenn Sie einen On-Premises-Server überwachen, wird dieses Feld nicht verwendet, und der Agent liest die Region aus dem `AmazonCloudWatchAgent`-Profil der AWS -Konfigurationsdatei.

- `credentials`— Gibt eine IAM-Rolle an, die beim Senden von Metriken, Protokollen und Traces an ein anderes AWS Konto verwendet werden soll. Sofern angegeben, enthält dieses Feld einen Parameter, `role_arn`.
  - `role_arn` – Gibt den Amazon Ressourcennamen (ARN) einer IAM-Rolle an, der für die Authentifizierung beim Senden von Metriken, Protokollen und Ablaufverfolgungen an ein anderes AWS -Konto verwendet werden soll. Weitere Informationen finden Sie unter Metriken, Protokolle und Ablaufverfolgungen an ein anderes Konto senden.
- `debug` – Optional. Gibt an, dass der CloudWatch Agent mit Debug-Protokollmeldungen ausgeführt wird. Der Standardwert ist `false`.
- `aws_sdk_log_level` – Optional. Wird nur in den Versionen 1.247350.0 und höher des Agenten unterstützt. CloudWatch

  Sie können dieses Feld angeben, damit der Agent die Protokollierung für AWS SDK-Endpunkte durchführt. Der Wert für dieses Feld kann eine oder mehrere der folgenden Optionen enthalten. Trennen Sie mehrere Optionen mit dem `|`-Zeichen.
  - `LogDebug`
  - `LogDebugWithSigning`
  - `LogDebugWithHTTPBody`
  - `LogDebugRequestRetries`
  - `LogDebugWithEventStreamBody`

  Weitere Informationen zu diesen Optionen finden Sie unter LogLevelType.
- `logfile`— Gibt den Ort an, an dem der CloudWatch Agent Protokollnachrichten schreibt. Wenn Sie eine leere Zeichenfolge angeben, wird das Protokoll in stderr abgelegt. Wenn Sie diese Option nicht angeben, lauten die Standardspeicherorte folgendermaßen:
  - Linux: `/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log`
  - Windows Server: `c:\\ProgramData\\Amazon\\CloudWatchAgent\\Logs\\amazon-cloudwatch-agent.log`

Der CloudWatch Agent rotiert die von ihm erstellte Protokolldatei automatisch. Eine Protokolldatei wird rotiert, wenn sie eine Größe von 100 MB erreicht. Der Agent bewahrt die rotierten Protokolldateien bis zu sieben Tage lang auf, und er behält bis zu fünf Backup-Protokolldateien, die ausgelagert werden. Die Dateinamen der Sicherungen von Protokolldateien werden um einen Zeitstempel ergänzt. Der Zeitstempel gibt das Datum und die Uhrzeit der Rotation an, z. B. `amazon-cloudwatch-agent-2018-06-08T21-01-50.247.log.gz`.

- `omit_hostname` – Optional. Standardmäßig wird der Hostname als Dimension von Metriken veröffentlicht, die vom Agenten erfasst werden, es sei denn, Sie verwenden das `append_dimensions`-Feld im `metrics`-Abschnitt. Setzen Sie `omit_hostname` auf `true`, um zu verhindern, dass der Hostname als Dimension veröffentlicht wird, auch wenn Sie `append_dimensions` nicht verwenden. Der Standardwert ist `false`.
- `run_as_user` – Optional. Gibt einen Benutzer an, der zum Ausführen des CloudWatch Agenten verwendet werden soll. Wenn Sie diesen Parameter nicht angeben, wird der Root-Benutzer verwendet. Diese Option ist nur auf Linux-Servern gültig.

  Wenn Sie diese Option angeben, muss der Benutzer vorhanden sein, bevor Sie den CloudWatch Agenten starten. Weitere Informationen finden Sie unter Den CloudWatch Agenten unter einem anderen Benutzer ausführen.
- `user_agent` – Optional. Gibt die `user-agent` Zeichenfolge an, die vom CloudWatch Agenten verwendet wird, wenn er API-Aufrufe an das CloudWatch Backend tätigt. Der Standardwert ist eine Zeichenfolge, die aus der Agentenversion, der Version der Go-Programmiersprache, die zum Kompilieren des Agenten verwendet wurde, dem Laufzeitbetriebssystem und der Architektur, der Buildzeit und den aktivierten Plugins besteht.
- `usage_data` Optional. Standardmäßig sendet der CloudWatch Agent Integritäts- und Leistungsdaten über sich selbst an, CloudWatch wann immer er Metriken oder Protokolle veröffentlicht. CloudWatch Für diese Daten entstehen Ihnen keine Kosten. Sie können verhindern, dass der Agent diese Daten sendet, indem Sie `false` für `usage_data` angeben. Wenn Sie diesen Parameter weglassen, wird der Standard von `true` verwendet, und der Agent sendet die Zustands- und Leistungsdaten.

  Wenn Sie diesen Wert auf `false` setzen, müssen Sie den Agenten beenden und neu starten, damit die Änderung wirksam wird.

Es folgt ein Beispiel für den Abschnitt `agent`.

```
"agent": {
```

```
   "metrics_collection_interval": 60,
   "region": "us-west-1",
   "logfile": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log",
   "debug": false,
   "run_as_user": "cwagent"
  }
```

CloudWatch Agenten-Konfigurationsdatei: Abschnitt „Metriken“

Felder, die Linux und Windows gemeinsam haben

Auf Servern mit Linux oder Windows Server enthält der Abschnitt `metrics` die folgenden Felder:

- `namespace` – Optional. Der Namespace für die vom Agent zu erfassenden Metriken. Der Standardwert ist `CWAgent`. Die maximale Länge beträgt 255 Zeichen. Im Folgenden wird ein Beispiel gezeigt:

  ```
  {
    "metrics": {
      "namespace": "Development/Product1Metrics",
     ......
     },
  }
  ```

- `append_dimensions` – Optional. Fügt Amazon-EC2-Metrik-Dimensionen zu allen vom Agent erfassten Metriken hinzu. Dies bewirkt auch, dass der Agent den Hostnamen nicht als Dimension veröffentlicht.

  Die einzigen unterstützten Schlüssel-Wert-Paare für `append_dimensions` werden in der folgenden Liste angezeigt. Alle anderen Schlüssel-Wert-Paare werden ignoriert. Der Agent unterstützt diese Schlüssel-Wert-Paare genau so, wie sie in der folgenden Liste aufgeführt sind. Sie können die Schlüsselwerte nicht ändern, um unterschiedliche Dimensionsnamen für sie zu veröffentlichen.

  - `"ImageId":"${aws:ImageId}"` legt die AMI-ID der Instance als Wert der `ImageId`-Dimension fest.
  - `"InstanceId":"${aws:InstanceId}"` legt die Instance-ID der Instance als Wert der `InstanceId`-Dimension fest.
  - `"InstanceType":"${aws:InstanceType}"` legt den Instance-Typ der Instance als Wert der `InstanceType`-Dimension fest.

- `"AutoScalingGroupName":"${aws:AutoScalingGroupName}"` legt den Namen der Auto Scaling-Gruppe der Instance als Wert der `AutoScalingGroupName`-Dimension fest.

Wenn Sie Dimensionen an Metriken mit beliebigen Schlüssel-Wert-Paaren anfügen möchten, verwenden Sie den Parameter `append_dimensions` im Feld für diesen bestimmten Metriktyp.

Wenn Sie einen Wert angeben, der von Amazon-EC2-Metadaten abhängt, und Sie Proxys verwenden, müssen Sie sicherstellen, dass der Server auf den Endpunkt für Amazon EC2 zugreifen kann. Weitere Informationen zu diesen Endpunkten finden Sie unter Amazon Elastic Compute Cloud (Amazon EC2) in der Allgemeine Amazon Web Services-Referenz.

- `aggregation_dimensions` – Optional. Gibt die Dimensionen an, in denen erfasste Metriken aggregiert werden sollen. Beispiel: Wenn Sie Metriken in der `AutoScalingGroupName`-Dimension zusammenführen, werden die Metriken aus allen Instances in der jeweiligen Auto-Scaling-Gruppe aggregiert und können als Ganzes angezeigt werden.

  Sie können Metriken in einer einzelnen oder in mehreren Dimensionen zusammenfassen. Beispiel: Wenn Sie `[["InstanceId"], ["InstanceType"], ["InstanceId","InstanceType"]]` festlegen, werden Metriken für die Instance-ID einzeln, den Instance-Typ einzeln und für die Kombination der beiden Dimensionen aggregiert.

  Sie können auch `[]` angeben, um alle Metriken in einer Sammlung ungeachtet jeglicher Dimensionen zusammenzufassen.

- `endpoint_override` – Gibt einen FIPS-Endpunkt oder einen privaten Link an, der als Endpunkt verwendet wird, an den der Agent Metriken sendet. Wenn Sie dies angeben und einen privaten Link einrichten, können Sie die Metriken an eine Amazon-VPC-Endpunkt senden. Weitere Informationen finden Sie unter Was ist Amazon VPC?

  Der Wert von `endpoint_override` muss eine Zeichenkette sein, die eine URL ist.

  Beispielsweise legt der folgende Teil des Metrikabschnitts der Konfigurationsdatei fest, dass der Agent beim Senden von Metriken einen VPC-Endpunkt verwendet.

```
{
  "metrics": {
    "endpoint_override": "vpce-XXXXXXXXXXXXXXXXXXXXXXXX.monitoring.us-
east-1.vpce.amazonaws.com",
   ......
   },
```

```
}
```

- `metrics_collected` – Erforderlich. Gibt an, welche Metriken erfasst werden sollen, einschließlich benutzerdefinierter Metriken, die mit `StatsD` oder `collectd` erfasst werden. Dieser Abschnitt enthält mehrere Unterabschnitte.

  Der Inhalt des `metrics_collected`-Abschnitts hängt davon ab, ob diese Konfigurationsdatei für einen Server mit Linux oder Windows Server vorgesehen ist.

- `force_flush_interval` – Gibt die maximale Zeitspanne in Sekunden an, in der Metriken im Speicherpuffer verbleiben, bevor sie an den Server gesendet werden. Unabhängig von dieser Einstellung werden die Metriken sofort an den Server gesendet, sobald die Größe der Metriken im Puffer 1 MB oder 1 000 verschiedene Metriken erreicht.

  Der Standardwert lautet 60.

- `credentials` – Gibt eine IAM-Rolle an, die beim Senden von Metriken an ein anderes Konto verwendet werden soll. Sofern angegeben, enthält dieses Feld einen Parameter, `role_arn`.
  - `role_arn` – Gibt den ARN einer IAM-Rolle für die Authentifizierung beim Senden von Metriken an ein anderes Konto an. Weitere Informationen finden Sie unter Metriken, Protokolle und Ablaufverfolgungen an ein anderes Konto senden. Wenn dieser Wert hier angegeben wird, überschreibt er den `role_arn` im Abschnitt `agent` der Konfigurationsdatei, sofern vorhanden.

Linux-Abschnitt

Auf Servern mit Linux kann der Abschnitt `metrics_collected` der Konfigurationsdatei auch die folgenden Felder enthalten.

Viele dieser Felder können `measurement`-Bereiche enthalten, in denen die Metriken aufgelistet werden, die Sie für diese Ressource erfassen möchten. In diesen `measurement`-Abschnitten können Sie entweder den vollständigen Metriknamen, wie z. B. `swap_used`, oder nur den Teil des Metriknamens, der an die Typ der Ressource angehängt wird. Beispiel: Die Angabe von `reads` im Abschnitt `measurement` des Abschnitts `diskio` bewirkt, dass die Metrik `diskio_reads` erfasst wird.

- `collectd` – Optional. Gibt an, dass Sie benutzerdefinierte Metriken mithilfe des Protokolls `collectd` abrufen möchten. Sie verwenden `collectd` Software, um die Metriken an den CloudWatch Agenten zu senden. Weitere Informationen zu den Konfigurationsoptionen, die für collectd verfügbar sind, finden Sie unter Abrufen benutzerdefinierter Metriken mit collectd.

- `cpu` – Optional. Gibt an, dass CPU-Metriken erfasst werden sollen. Dieser Abschnitt gilt nur für Linux-Instances. Sie müssen mindestens eines der `resources`- und `totalcpu`-Felder für alle zu erfassenden CPU-Metriken einschließen. Dieser Abschnitt kann die folgenden Felder enthalten:
  - `drop_original_metrics` – Optional. Wenn Sie das `aggregation_dimensions`-Feld im `metrics`-Abschnitt verwenden, um Metriken zu aggregierten Ergebnissen zusammenzufassen, dann sendet der Agent standardmäßig sowohl die aggregierten Metriken als auch die ursprünglichen Metriken, die für jeden Wert der Dimension getrennt sind. Wenn Sie nicht möchten, dass die ursprünglichen Metriken an gesendet werden CloudWatch, können Sie diesen Parameter mit einer Liste von Metriken angeben. Für die zusammen mit diesem Parameter angegebenen Metriken werden keine Kennzahlen nach Dimension gemeldet CloudWatch. Stattdessen werden nur die aggregierten Metriken gemeldet. Dadurch verringert sich die Anzahl der Metriken, die der Agent erfasst, was Ihre Kosten senkt.
  - `resources` Optional. Geben Sie dieses Feld mit dem Wert `*` an, um zu bewirken, dass pro CPU-Metriken erfasst werden. Der einzige zulässige Wert ist `*`.
  - `totalcpu` – Optional. Gibt an, ob CPU-Metriken gesammelt über alle CPU-Kerne hinweg gemeldet werden sollen. Der Standardwert ist „true“.
  - `measurement` – Gibt das Array der zu erfassenden CPU-Metriken an. Mögliche Werte sind `time_active`, `time_guest`, `time_guest_nice`, `time_idle`, `time_iowait`, `time_irq`, `time_nice`, `time_softirq`, `time_steal`, `time_system`, `time_user`, `usage_active`, `usage_guest`, `usage_guest_nice`, `usage_idle`, `usage_iowait`, `usage_irq`, `usage_nice`, `usage_softirq`, `usage_steal`, `usage_system` und `usage_user`. Dieses Feld muss angegeben werden, wenn Sie `cpu` einbeziehen.

    Standardmäßig ist die Einheit für `cpu_usage_*`-Metriken `Percent`; `cpu_time_*`-Metriken sind einheitenlos.

    Im Eintrag für jede einzelne Metrik können Sie optional einen oder beide der folgenden Werte angeben:
    - `rename` – Legt einen anderen Namen für diese Metrik fest.
    - `unit` – Gibt die zu verwendende Einheit für diese Metrik an und überschreibt die Standardeinheit für die Metrik (`None`). Bei der von Ihnen angegebenen Einheit muss es sich um eine gültige CloudWatch metrische Einheit handeln, wie in der `Unit` Beschreibung unter aufgeführt MetricDatum.
  - `metrics_collection_interval` – Optional. Gibt an, wie oft die CPU-Metriken erfasst werden und das globale `metrics_collection_interval` im Abschnitt `agent` der Konfigurationsdatei überschrieben wird.

Der Wert wird in Sekunden angegeben. Beispiel: Angeben, dass 10 Metriken alle 10 Sekunden und 300 Metriken alle 5 Minuten gesammelt werden sollen.

Wenn Sie diesen Wert auf weniger als 60 Sekunden festlegen, wird die jeweilige Metrik als hochauflösende Metrik erfasst. Weitere Informationen zu hochauflösenden Metriken finden Sie unter Hochauflösende Metriken.

- `append_dimensions` – Optional. Zusätzliche Dimensionen, die nur für die CPU-Metriken verwendet werden sollen. Falls Sie dieses Feld angeben, wird es zusätzlich zu den im globalen Feld `append_dimensions` angegebenen Dimensionen verwendet, das für alle Typen von Metriken verwendet wird, die vom Agenten erfasst werden.

- `disk` – Optional. Gibt an, dass Datenträger-Metriken erfasst werden sollen. Dieser Abschnitt gilt nur für Linux-Instances. Dieser Abschnitt kann die folgenden Felder enthalten:
  - `drop_original_metrics` – Optional. Wenn Sie das `aggregation_dimensions`-Feld im `metrics`-Abschnitt verwenden, um Metriken zu aggregierten Ergebnissen zusammenzufassen, dann sendet der Agent standardmäßig sowohl die aggregierten Metriken als auch die ursprünglichen Metriken, die für jeden Wert der Dimension getrennt sind. Wenn Sie nicht möchten, dass die ursprünglichen Messwerte gesendet werden CloudWatch, können Sie diesen Parameter mit einer Liste von Metriken angeben. Für die zusammen mit diesem Parameter angegebenen Metriken werden keine Kennzahlen nach Dimension gemeldet CloudWatch. Stattdessen werden nur die aggregierten Metriken gemeldet. Dadurch verringert sich die Anzahl der Metriken, die der Agent erfasst, was Ihre Kosten senkt.
  - `resources` – Optional. Gibt ein Array von Datenträger-Mountingpunkten an. Dieses Feld beschränkt CloudWatch sich darauf, nur Metriken von den aufgelisteten Einhängepunkten zu sammeln. Sie können `*` als Wert festlegen, um Metriken von allen Mountingpunkten zu erfassen. Standardmäßig werden Metriken von allen Mountingpunkten erfasst.
  - `measurement` – Gibt das Array der zu erfassenden Disk-Metriken an. Mögliche Werte sind `free`, `total`, `used`, `used_percent`, `inodes_free`, `inodes_used` und `inodes_total`. Dieses Feld muss angegeben werden, wenn Sie `disk` einbeziehen.

> **Note**
>
> Die `disk`-Metriken haben eine Dimension für `Partition`, was bedeutet, dass die Anzahl der generierten benutzerdefinierten Metriken von der Anzahl der Partitionen abhängt, die Ihrer Instance zugeordnet sind. Die Anzahl der Festplattenpartitionen hängt

davon ab, welches AMI Sie verwenden, und wie viele Amazon-EBS-Volumes Sie an den Server anfügen.

Informationen zum Anzeigen der Standardeinheiten für jede `disk`-Metrik finden Sie unter Vom CloudWatch Agenten auf Linux- und macOS-Instances gesammelte Metriken.

Im Eintrag für jede einzelne Metrik können Sie optional einen oder beide der folgenden Werte angeben:

- `rename` – Legt einen anderen Namen für diese Metrik fest.
- `unit` – Gibt die zu verwendende Einheit für diese Metrik an und überschreibt die Standardeinheit für die Metrik (`None` von `None`). Bei der von Ihnen angegebenen Einheit muss es sich um eine gültige CloudWatch metrische Einheit handeln, wie in der `Unit` Beschreibung unter aufgeführt MetricDatum.

- `ignore_file_system_types` – Gibt Dateisystemtypen an, die beim Erfassen von Datenträgermetriken ausgeschlossen werden sollen. Gültige Werte sind `sysfs`, `devtmpfs` usw.
- `drop_device` – Wenn Sie dies auf `true` setzen, wird `Device` nicht als Dimension für Datenträgermetriken aufgenommen.

  Verhindern, dass `Device` als Dimension verwendet wird, kann auf Instances nützlich sein, die das Nitro-System verwenden, da sich die Gerätenamen auf diesen Instances bei jedem Datenträger-Mount ändern, wenn die Instance neu gestartet wird. Dies kann dazu führen, dass inkonsistente Daten in Ihren -Metriken und dazu führen, dass Alarme, die auf diesen Metriken basieren, in den Status `INSUFFICIENT DATA` übergehen.

  Der Standardwert ist `false`.
- `metrics_collection_interval` – Optional. Gibt an, wie oft die Datenträger-Metriken erfasst werden und das globale `metrics_collection_interval` im Abschnitt `agent` der Konfigurationsdatei überschrieben wird.

  Der Wert wird in Sekunden angegeben.

  Wenn Sie diesen Wert auf weniger als 60 Sekunden festlegen, wird die jeweilige Metrik als hochauflösende Metrik erfasst. Weitere Informationen finden Sie unter Hochauflösende Metriken.
- `append_dimensions` – Optional. Zusätzliche Dimensionen, die nur für die Datenträger-Metriken verwendet werden sollen. Falls Sie dieses Feld angeben, wird es zusätzlich zu den

im `append_dimensions`-Feld angegebenen Dimensionen verwendet, das für alle Typen von Metriken verwendet wird, die vom Agent erfasst werden.

- `diskio` – Optional. Gibt an, dass Diskio-Metriken erfasst werden sollen. Dieser Abschnitt gilt nur für Linux-Instances. Dieser Abschnitt kann die folgenden Felder enthalten:
  - `drop_original_metrics` – Optional. Wenn Sie das `aggregation_dimensions`-Feld im `metrics`-Abschnitt verwenden, um Metriken zu aggregierten Ergebnissen zusammenzufassen, dann sendet der Agent standardmäßig sowohl die aggregierten Metriken als auch die ursprünglichen Metriken, die für jeden Wert der Dimension getrennt sind. Wenn Sie nicht möchten, dass die ursprünglichen Messwerte gesendet werden CloudWatch, können Sie diesen Parameter mit einer Liste von Metriken angeben. Für die zusammen mit diesem Parameter angegebenen Metriken werden keine Kennzahlen nach Dimension gemeldet CloudWatch. Stattdessen werden nur die aggregierten Metriken gemeldet. Dadurch verringert sich die Anzahl der Metriken, die der Agent erfasst, was Ihre Kosten senkt.
  - `resources` Optional. Wenn Sie eine Reihe von Geräten angeben, werden nur Messwerte von diesen Geräten CloudWatch erfasst. Andernfalls werden Metriken für alle Geräte erfasst. Sie können auch * als Wert festlegen, um Metriken von allen Geräten zu erfassen.
  - `measurement` – Gibt das Array der zu erfassenden Diskio-Metriken an. Mögliche Werte sind `reads`, `writes`, `read_bytes`, `write_bytes`, `read_time`, `write_time`, `io_time` und `iops_in_progress`. Dieses Feld muss angegeben werden, wenn Sie `diskio` einbeziehen.

    Im Eintrag für jede einzelne Metrik können Sie optional einen oder beide der folgenden Werte angeben:
    - `rename` – Legt einen anderen Namen für diese Metrik fest.
    - `unit` – Gibt die zu verwendende Einheit für diese Metrik an und überschreibt die Standardeinheit für die Metrik (`None` von `None`). Bei der von Ihnen angegebenen Einheit muss es sich um eine gültige CloudWatch metrische Einheit handeln, wie in der `Unit` Beschreibung unter aufgeführt MetricDatum.
  - `metrics_collection_interval` – Optional. Gibt an, wie oft die diskio-Metriken erfasst werden und das globale `metrics_collection_interval` im Abschnitt `agent` der Konfigurationsdatei überschrieben wird.

    Der Wert wird in Sekunden angegeben.

    Wenn Sie diesen Wert auf weniger als 60 Sekunden festlegen, wird die jeweilige Metrik als hochauflösende Metrik erfasst. Weitere Informationen zu hochauflösenden Metriken finden Sie unter Hochauflösende Metriken.

  - `append_dimensions` – Optional. Zusätzliche Dimensionen, die nur für die diskio-Metriken verwendet werden sollen. Falls Sie dieses Feld angeben, wird es zusätzlich zu den im `append_dimensions`-Feld angegebenen Dimensionen verwendet, das für alle Typen von Metriken verwendet wird, die vom Agent erfasst werden.

- `swap` – Optional. Gibt an, dass Swap-Arbeitsspeicher-Metriken erfasst werden sollen. Dieser Abschnitt gilt nur für Linux-Instances. Dieser Abschnitt kann die folgenden Felder enthalten:
  - `drop_original_metrics` – Optional. Wenn Sie das `aggregation_dimensions`-Feld im `metrics`-Abschnitt verwenden, um Metriken zu aggregierten Ergebnissen zusammenzufassen, dann sendet der Agent standardmäßig sowohl die aggregierten Metriken als auch die ursprünglichen Metriken, die für jeden Wert der Dimension getrennt sind. Wenn Sie nicht möchten, dass die ursprünglichen Messwerte gesendet werden CloudWatch, können Sie diesen Parameter mit einer Liste von Metriken angeben. Für die zusammen mit diesem Parameter angegebenen Metriken werden keine Kennzahlen nach Dimension gemeldet CloudWatch. Stattdessen werden nur die aggregierten Metriken gemeldet. Dadurch verringert sich die Anzahl der Metriken, die der Agent erfasst, was Ihre Kosten senkt.
  - `measurement` – Gibt das Array der zu erfassenden Swap-Metriken an. Mögliche Werte sind `free`, `used` und `used_percent`. Dieses Feld muss angegeben werden, wenn Sie `swap` einbeziehen.

    Informationen zum Anzeigen der Standardeinheiten für jede `swap`-Metrik finden Sie unter Vom CloudWatch Agenten auf Linux- und macOS-Instances gesammelte Metriken.

    Im Eintrag für jede einzelne Metrik können Sie optional einen oder beide der folgenden Werte angeben:

    - `rename` – Legt einen anderen Namen für diese Metrik fest.
    - `unit` – Gibt die zu verwendende Einheit für diese Metrik an und überschreibt die Standardeinheit für die Metrik (`None` von `None`). Bei der von Ihnen angegebenen Einheit muss es sich um eine gültige CloudWatch metrische Einheit handeln, wie in der `Unit` Beschreibung unter aufgeführt MetricDatum.
  - `metrics_collection_interval` – Optional. Gibt an, wie oft die Swap-Metriken erfasst werden und das globale `metrics_collection_interval` im Abschnitt `agent` der Konfigurationsdatei überschrieben wird.

    Der Wert wird in Sekunden angegeben.

Wenn Sie diesen Wert auf weniger als 60 Sekunden festlegen, wird die jeweilige Metrik als hochauflösende Metrik erfasst. Weitere Informationen zu hochauflösenden Metriken finden Sie unter Hochauflösende Metriken.

- `append_dimensions` – Optional. Zusätzliche Dimensionen, die nur für die Swap-Metriken verwendet werden sollen. Falls Sie dieses Feld angeben, wird es zusätzlich zu den im globalen `append_dimensions`-Feld angegebenen Dimensionen verwendet, das für alle Typen von Metriken verwendet wird, die vom Agent erfasst werden. Der Wert wird als hochauflösende Metrik erfasst.

- `mem` – Optional. Gibt an, dass Arbeitsspeicher-Metriken erfasst werden sollen. Dieser Abschnitt gilt nur für Linux-Instances. Dieser Abschnitt kann die folgenden Felder enthalten:
  - `drop_original_metrics` – Optional. Wenn Sie das `aggregation_dimensions`-Feld im `metrics`-Abschnitt verwenden, um Metriken zu aggregierten Ergebnissen zusammenzufassen, dann sendet der Agent standardmäßig sowohl die aggregierten Metriken als auch die ursprünglichen Metriken, die für jeden Wert der Dimension getrennt sind. Wenn Sie nicht möchten, dass die ursprünglichen Messwerte gesendet werden CloudWatch, können Sie diesen Parameter mit einer Liste von Metriken angeben. Für die zusammen mit diesem Parameter angegebenen Metriken werden keine Kennzahlen nach Dimension gemeldet CloudWatch. Stattdessen werden nur die aggregierten Metriken gemeldet. Dadurch verringert sich die Anzahl der Metriken, die der Agent erfasst, was Ihre Kosten senkt.
  - `measurement` – Gibt das Array der zu erfassenden Arbeitsspeicher-Metriken an. Mögliche Werte sind `active`, `available`, `available_percent`, `buffered`, `cached`, `free`, `inactive`, `total`, `used` und `used_percent`. Dieses Feld muss angegeben werden, wenn Sie `mem` einbeziehen.

    Informationen zum Anzeigen der Standardeinheiten für jede `mem`-Metrik finden Sie unter Vom CloudWatch Agenten auf Linux- und macOS-Instances gesammelte Metriken.

    Im Eintrag für jede einzelne Metrik können Sie optional einen oder beide der folgenden Werte angeben:
    - `rename` – Legt einen anderen Namen für diese Metrik fest.
    - `unit` – Gibt die zu verwendende Einheit für diese Metrik an und überschreibt die Standardeinheit für die Metrik (`None`). Bei der von Ihnen angegebenen Einheit muss es sich um eine gültige CloudWatch metrische Einheit handeln, wie in der `Unit` Beschreibung unter aufgeführt MetricDatum.

- `metrics_collection_interval` – Optional. Gibt an, wie oft die mem-Metriken erfasst werden und das globale `metrics_collection_interval` im Abschnitt `agent` der Konfigurationsdatei überschrieben wird.

  Der Wert wird in Sekunden angegeben.

  Wenn Sie diesen Wert auf weniger als 60 Sekunden festlegen, wird die jeweilige Metrik als hochauflösende Metrik erfasst. Weitere Informationen zu hochauflösenden Metriken finden Sie unter Hochauflösende Metriken.

- `append_dimensions` – Optional. Zusätzliche Dimensionen, die nur für die mem-Metriken verwendet werden sollen. Falls Sie dieses Feld angeben, wird es zusätzlich zu den im Feld `append_dimensions` angegebenen Dimensionen verwendet, das für alle Typen von Metriken verwendet wird, die vom Agenten erfasst werden.

- `net` – Optional. Gibt an, dass Netzwerk-Metriken erfasst werden sollen. Dieser Abschnitt gilt nur für Linux-Instances. Dieser Abschnitt kann die folgenden Felder enthalten:

  - `drop_original_metrics` – Optional. Wenn Sie das `aggregation_dimensions`-Feld im `metrics`-Abschnitt verwenden, um Metriken zu aggregierten Ergebnissen zusammenzufassen, dann sendet der Agent standardmäßig sowohl die aggregierten Metriken als auch die ursprünglichen Metriken, die für jeden Wert der Dimension getrennt sind. Wenn Sie nicht möchten, dass die ursprünglichen Messwerte gesendet werden CloudWatch, können Sie diesen Parameter mit einer Liste von Metriken angeben. Für die zusammen mit diesem Parameter angegebenen Metriken werden keine Kennzahlen nach Dimension gemeldet CloudWatch. Stattdessen werden nur die aggregierten Metriken gemeldet. Dadurch verringert sich die Anzahl der Metriken, die der Agent erfasst, was Ihre Kosten senkt.

  - `resources` Optional. Wenn Sie ein Array von Netzwerkschnittstellen angeben, werden nur Metriken von diesen Schnittstellen CloudWatch erfasst. Andernfalls werden Metriken für alle Geräte erfasst. Sie können auch * als Wert festlegen, um Metriken von allen Schnittstellen zu erfassen.

  - `measurement` – Gibt das Array der zu erfassenden Netzwerk-Metriken an. Mögliche Werte sind `bytes_sent`, `bytes_recv`, `drop_in`, `drop_out`, `err_in`, `err_out`, `packets_sent` und `packets_recv`. Dieses Feld muss angegeben werden, wenn Sie `net` einbeziehen.

    Informationen zum Anzeigen der Standardeinheiten für jede `net`-Metrik finden Sie unter Vom CloudWatch Agenten auf Linux- und macOS-Instances gesammelte Metriken.

Im Eintrag für jede einzelne Metrik können Sie optional einen oder beide der folgenden Werte angeben:

- `rename` – Legt einen anderen Namen für diese Metrik fest.
- `unit` – Gibt die zu verwendende Einheit für diese Metrik an und überschreibt die Standardeinheit für die Metrik (`None`). Bei der von Ihnen angegebenen Einheit muss es sich um eine gültige CloudWatch metrische Einheit handeln, wie in der `Unit` Beschreibung unter aufgeführt MetricDatum.

- `metrics_collection_interval` – Optional. Gibt an, wie oft die net-Metriken erfasst werden und das globale `metrics_collection_interval` im Abschnitt `agent` der Konfigurationsdatei überschrieben wird.

  Der Wert wird in Sekunden angegeben. Beispiel: Angeben, dass 10 Metriken alle 10 Sekunden und 300 Metriken alle 5 Minuten gesammelt werden sollen.

  Wenn Sie diesen Wert auf weniger als 60 Sekunden festlegen, wird die jeweilige Metrik als hochauflösende Metrik erfasst. Weitere Informationen zu hochauflösenden Metriken finden Sie unter Hochauflösende Metriken.

- `append_dimensions` – Optional. Zusätzliche Dimensionen, die nur für die net-Metriken verwendet werden sollen. Falls Sie dieses Feld angeben, wird es zusätzlich zu den im Feld `append_dimensions` angegebenen Dimensionen verwendet, das für alle Typen von Metriken verwendet wird, die vom Agent erfasst werden.

- `netstat` – Optional. Gibt an, dass TCP-Verbindungsstatus und UDP-Verbindungsmetriken gesammelt werden sollen. Dieser Abschnitt gilt nur für Linux-Instances. Dieser Abschnitt kann die folgenden Felder enthalten:
  - `drop_original_metrics` – Optional. Wenn Sie das `aggregation_dimensions`-Feld im `metrics`-Abschnitt verwenden, um Metriken zu aggregierten Ergebnissen zusammenzufassen, dann sendet der Agent standardmäßig sowohl die aggregierten Metriken als auch die ursprünglichen Metriken, die für jeden Wert der Dimension getrennt sind. Wenn Sie nicht möchten, dass die ursprünglichen Messwerte gesendet werden CloudWatch, können Sie diesen Parameter mit einer Liste von Metriken angeben. Für die zusammen mit diesem Parameter angegebenen Metriken werden keine Kennzahlen nach Dimension gemeldet CloudWatch. Stattdessen werden nur die aggregierten Metriken gemeldet. Dadurch verringert sich die Anzahl der Metriken, die der Agent erfasst, was Ihre Kosten senkt.
  - `measurement` – Gibt das Array der zu erfassenden Netstat-Metriken an. Mögliche Werte sind `tcp_close`, `tcp_close_wait`, `tcp_closing`, `tcp_established`, `tcp_fin_wait1`,

`tcp_fin_wait2`, `tcp_last_ack`, `tcp_listen`, `tcp_none`, `tcp_syn_sent`, `tcp_syn_recv`, `tcp_time_wait` und `udp_socket`. Dieses Feld muss angegeben werden, wenn Sie `netstat` einbeziehen.

Informationen zum Anzeigen der Standardeinheiten für jede `netstat`-Metrik finden Sie unter Vom CloudWatch Agenten auf Linux- und macOS-Instances gesammelte Metriken.

Im Eintrag für jede einzelne Metrik können Sie optional einen oder beide der folgenden Werte angeben:

- `rename` – Legt einen anderen Namen für diese Metrik fest.
- `unit` – Gibt die zu verwendende Einheit für diese Metrik an und überschreibt die Standardeinheit für die Metrik (`None`). Bei der von Ihnen angegebenen Einheit muss es sich um eine gültige CloudWatch metrische Einheit handeln, wie in der `Unit` Beschreibung unter aufgeführt MetricDatum.

- `metrics_collection_interval` – Optional. Gibt an, wie oft die netstat-Metriken erfasst werden und das globale `metrics_collection_interval` im Abschnitt `agent` der Konfigurationsdatei überschrieben wird.

  Der Wert wird in Sekunden angegeben.

  Wenn Sie diesen Wert auf weniger als 60 Sekunden festlegen, wird die jeweilige Metrik als hochauflösende Metrik erfasst. Weitere Informationen zu hochauflösenden Metriken finden Sie unter Hochauflösende Metriken.

- `append_dimensions` – Optional. Zusätzliche Dimensionen, die nur für die netstat-Metriken verwendet werden sollen. Falls Sie dieses Feld angeben, wird es zusätzlich zu den im Feld `append_dimensions` angegebenen Dimensionen verwendet, das für alle Typen von Metriken verwendet wird, die vom Agent erfasst werden.

- `processes` – Optional. Gibt an, dass Prozess-Metriken erfasst werden sollen. Dieser Abschnitt gilt nur für Linux-Instances. Dieser Abschnitt kann die folgenden Felder enthalten:

  - `drop_original_metrics` – Optional. Wenn Sie das `aggregation_dimensions`-Feld im `metrics`-Abschnitt verwenden, um Metriken zu aggregierten Ergebnissen zusammenzufassen, dann sendet der Agent standardmäßig sowohl die aggregierten Metriken als auch die ursprünglichen Metriken, die für jeden Wert der Dimension getrennt sind. Wenn Sie nicht möchten, dass die ursprünglichen Messwerte gesendet werden CloudWatch, können Sie diesen Parameter mit einer Liste von Metriken angeben. Für die zusammen mit diesem Parameter angegebenen Metriken werden keine Kennzahlen nach Dimension gemeldet CloudWatch.

Stattdessen werden nur die aggregierten Metriken gemeldet. Dadurch verringert sich die Anzahl der Metriken, die der Agent erfasst, was Ihre Kosten senkt.

- `measurement` – Gibt das Array der zu erfassenden Prozess-Metriken an. Mögliche Werte sind `blocked`, `dead`, `idle`, `paging`, `running`, `sleeping`, `stopped`, `total`, `total_threads`, `wait` und `zombies`. Dieses Feld muss angegeben werden, wenn Sie `processes` einbeziehen.

  Für alle `processes`-Metriken lautet die Standardeinheit `None`.

  Im Eintrag für jede einzelne Metrik können Sie optional einen oder beide der folgenden Werte angeben:

  - `rename` – Legt einen anderen Namen für diese Metrik fest.
  - `unit` – Gibt die zu verwendende Einheit für diese Metrik an und überschreibt die Standardeinheit für die Metrik (`None`). Bei der von Ihnen angegebenen Einheit muss es sich um eine gültige CloudWatch metrische Einheit handeln, wie in der `Unit` Beschreibung unter aufgeführt MetricDatum.
- `metrics_collection_interval` – Optional. Gibt an, wie oft die Prozess-Metriken erfasst werden und das globale `metrics_collection_interval` im Abschnitt `agent` der Konfigurationsdatei überschrieben wird.

  Der Wert wird in Sekunden angegeben. Beispiel: Angeben, dass 10 Metriken alle 10 Sekunden und 300 Metriken alle 5 Minuten gesammelt werden sollen.

  Wenn Sie diesen Wert auf weniger als 60 Sekunden festlegen, wird die jeweilige Metrik als hochauflösende Metrik erfasst. Weitere Informationen finden Sie unter Hochauflösende Metriken.
- `append_dimensions` – Optional. Zusätzliche Dimensionen, die nur für die Prozess-Metriken verwendet werden sollen. Falls Sie dieses Feld angeben, wird es zusätzlich zu den im Feld `append_dimensions` angegebenen Dimensionen verwendet, das für alle Typen von Metriken verwendet wird, die vom Agent erfasst werden.

- `nvidia_gpu` – Optional. Gibt an, dass NVIDIA GPU-Metriken erfasst werden sollen. Dieser Abschnitt gilt nur für Linux-Instances auf Hosts, die mit einem NVIDIA GPU-Accelerator konfiguriert sind und auf denen das NVIDIA System Management Interface (nvidia-smi) installiert ist.

  Den erfassten NVIDIA GPU-Metriken wird die Zeichenfolge `nvidia_smi_` vorangestellt, um sie von den Metriken zu unterscheiden, die für andere Accelerator-Typen erfasst wurden. Dieser Abschnitt kann die folgenden Felder enthalten:

  - `drop_original_metrics` – Optional. Wenn Sie das `aggregation_dimensions`-Feld im `metrics`-Abschnitt verwenden, um Metriken zu aggregierten Ergebnissen zusammenzufassen,

dann sendet der Agent standardmäßig sowohl die aggregierten Metriken als auch die ursprünglichen Metriken, die für jeden Wert der Dimension getrennt sind. Wenn Sie nicht möchten, dass die ursprünglichen Messwerte gesendet werden CloudWatch, können Sie diesen Parameter mit einer Liste von Metriken angeben. Für die zusammen mit diesem Parameter angegebenen Metriken werden keine Kennzahlen nach Dimension gemeldet CloudWatch. Stattdessen werden nur die aggregierten Metriken gemeldet. Dadurch verringert sich die Anzahl der Metriken, die der Agent erfasst, was Ihre Kosten senkt.

- `measurement` – Gibt das Array der zu erfassenden NVIDIA GPU-Metriken an. Eine Liste der möglichen Werte, die Sie hier verwenden können, finden Sie in der Spalte Metric (Metrik) in der Tabelle unter Erfassen von NVIDIA GPU-Metriken.

  Im Eintrag für jede einzelne Metrik können Sie optional einen oder beide der folgenden Werte angeben:

  - `rename` – Legt einen anderen Namen für diese Metrik fest.
  - `unit` – Gibt die zu verwendende Einheit für diese Metrik an und überschreibt die Standardeinheit für die Metrik (`None`). Bei der von Ihnen angegebenen Einheit muss es sich um eine gültige CloudWatch metrische Einheit handeln, wie in der `Unit` Beschreibung unter aufgeführt MetricDatum.

- `metrics_collection_interval` Optional. Gibt an, wie oft die NVIDIA GPU-Metriken erfasst werden und das globale `metrics_collection_interval` im Abschnitt `agent` der Konfigurationsdatei überschrieben wird.

- `procstat` – Optional. Gibt an, dass Sie Metriken aus einzelnen Prozessen abrufen möchten. Weitere Informationen zu den Konfigurationsoptionen, die für procstat verfügbar sind, finden Sie unter Erfassen von Prozessmetriken mit dem procstat-Plugin.
- `statsd` – Optional. Gibt an, dass Sie benutzerdefinierte Metriken mithilfe des Protokolls `StatsD` abrufen möchten. Der CloudWatch Agent fungiert als Daemon für das Protokoll. Sie verwenden einen beliebigen `StatsD` Standard-Client, um die Metriken an den CloudWatch Agenten zu senden. Weitere Informationen zu den Konfigurationsoptionen, die für StatsD verfügbar sind, finden Sie unter Abrufen benutzerdefinierter Metriken mit StatsD .
- `ethtool` – Optional. Gibt an, dass Sie Netzwerkmetriken mithilfe des `ethtool`-Plug-Ins abrufen möchten. Dieses Plugin kann sowohl die vom Standard-Dienstprogramm ethtool gesammelten Metriken als auch die Metriken zur Netzwerkleistung von Amazon-EC2-Instances importieren. Weitere Informationen zu den Konfigurationsoptionen, die für ethtool verfügbar sind, finden Sie unter Netzwerkleistungsmetriken sammeln.

Es folgt das Beispiel eines `metrics`-Abschnitts für einen Linux-Server. In diesem Beispiel werden drei CPU-Metriken, drei netstat-Metriken, drei Prozessmetriken und eine Datenträgermetrik erfasst und der Agent ist zum Empfang zusätzlicher Metriken von einem `collectd`-Client eingerichtet.

```
"metrics": {
    "aggregation_dimensions" : [["AutoScalingGroupName"], ["InstanceId",
 "InstanceType"],[]],
    "metrics_collected": {
      "collectd": {},
      "cpu": {
        "resources": [
          "*"
        ],
        "measurement": [
          {"name": "cpu_usage_idle", "rename": "CPU_USAGE_IDLE", "unit": "Percent"},
          {"name": "cpu_usage_nice", "unit": "Percent"},
          "cpu_usage_guest"
        ],
        "totalcpu": false,
        "drop_original_metrics": [ "cpu_usage_guest" ],
        "metrics_collection_interval": 10,
        "append_dimensions": {
          "test": "test1",
          "date": "2017-10-01"
        }
      },
      "netstat": {
        "measurement": [
          "tcp_established",
          "tcp_syn_sent",
          "tcp_close"
        ],
        "metrics_collection_interval": 60
      },
       "disk": {
        "measurement": [
          "used_percent"
        ],
        "resources": [
          "*"
        ],
        "drop_device": true
      },
```

```
      "processes": {
        "measurement": [
          "running",
          "sleeping",
          "dead"
        ]
      }
    },
    "append_dimensions": {
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}",
      "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
    }
  }
```

Windows Server

Im Abschnitt `metrics_collected` für Windows Server können Sie für jedes Windows-Leistungsobjekt Unterabschnitte anlegen, beispielsweise `Memory`, `Processor` und `LogicalDisk`. Informationen darüber, welche Objekte und Zähler verfügbar sind, finden Sie unter Leistungszähler in der Microsoft Windows-Dokumentation.

Innerhalb des Unterabschnitts für jedes Objekt geben Sie ein `measurement`-Array der zu erfassenden Zähler an. Das `measurement`-Array ist für jedes Objekt erforderlich, das Sie in der Konfigurationsdatei angeben. Sie können auch ein `resources`-Feld angeben, um die Instances zu benennen, aus denen Sie Metriken erfassen. Sie können auch `*` für `resources` angeben, um separate Metriken für alle Instances zu erfassen. Wenn Sie `resources` bei Zählern mit Instances weglassen, werden die Daten für alle Instances in einem Satz zusammengefasst. Wenn Sie `resources` bei Zählern, die keine Instanzen haben, auslassen, werden die Zähler nicht vom Agenten erfasst. CloudWatch Um festzustellen, ob Zähler über Instances verfügen, können Sie einen der folgenden Befehle verwenden.

Powershell:

```
Get-Counter -ListSet *
```

Befehlszeile (nicht Powershell):

```
TypePerf.exe -q
```

Innerhalb des jeweiligen Objektabschnitts können Sie auch die folgenden optionalen Felder angeben:

- `metrics_collection_interval` – Optional. Gibt an, wie oft die Metriken für dieses Objekt erfasst werden und das globale `metrics_collection_interval` im Abschnitt `agent` der Konfigurationsdatei überschrieben wird.

  Der Wert wird in Sekunden angegeben. Beispiel: Angeben, dass 10 Metriken alle 10 Sekunden und 300 Metriken alle 5 Minuten gesammelt werden sollen.

  Wenn Sie diesen Wert auf weniger als 60 Sekunden festlegen, wird die jeweilige Metrik als hochauflösende Metrik erfasst. Weitere Informationen finden Sie unter Hochauflösende Metriken.
- `append_dimensions` – Optional. Gibt zusätzliche Dimensionen an, die nur für die Metriken für dieses Objekt verwendet werden sollen. Falls Sie dieses Feld angeben, wird es zusätzlich zu den im globalen Feld `append_dimensions` angegebenen Dimensionen verwendet, das für alle Typen von Metriken verwendet wird, die vom Agent erfasst werden.
- `drop_original_metrics` Optional. Wenn Sie das `aggregation_dimensions`-Feld im `metrics`-Abschnitt verwenden, um Metriken zu aggregierten Ergebnissen zusammenzufassen, dann sendet der Agent standardmäßig sowohl die aggregierten Metriken als auch die ursprünglichen Metriken, die für jeden Wert der Dimension getrennt sind. Wenn Sie nicht möchten, dass die ursprünglichen Metriken an gesendet werden CloudWatch, können Sie diesen Parameter mit einer Liste von Metriken angeben. Für die zusammen mit diesem Parameter angegebenen Metriken werden keine Kennzahlen nach Dimension gemeldet CloudWatch. Stattdessen werden nur die aggregierten Metriken gemeldet. Dadurch verringert sich die Anzahl der Metriken, die der Agent erfasst, was Ihre Kosten senkt.

Innerhalb des jeweiligen Zählerabschnitts können Sie auch die folgenden optionalen Felder angeben:

- `rename`— Gibt einen anderen Namen an, der CloudWatch für diese Metrik verwendet werden soll.
- `unit` – Gibt die für diese Metrik zu verwendende Einheit an. Bei der von Ihnen angegebenen Einheit muss es sich um eine gültige CloudWatch metrische Einheit handeln, wie in der `Unit` Beschreibung unter aufgeführt MetricDatum.

Es gibt zwei weitere optionale Abschnitte, die Sie in `metrics_collected` aufnehmen können:

- `statsd` – Ermöglicht den Abruf benutzerdefinierter Metriken mittels `StatsD`-Protokoll. Der CloudWatch Agent fungiert als Daemon für das Protokoll. Sie verwenden einen beliebigen `StatsD`

Standard-Client, um die Metriken an den CloudWatch Agenten zu senden. Weitere Informationen finden Sie unter Abrufen benutzerdefinierter Metriken mit StatsD .

- `procstat` – Ermöglicht den Abruf von Metriken aus einzelnen Prozessen. Weitere Informationen finden Sie unter Erfassen von Prozessmetriken mit dem procstat-Plugin.

Es folgt das Beispiel eines `metrics`-Abschnitts zur Verwendung unter Windows Server. In diesem Beispiel werden viele Windows-Metriken erfasst, und der Computer ist zusätzlich für das Empfangen weiterer Metriken von einem `StatsD`-Client eingerichtet.

```
"metrics": {
    "metrics_collected": {
      "statsd": {},
      "Processor": {
        "measurement": [
          {"name": "% Idle Time", "rename": "CPU_IDLE", "unit": "Percent"},
          "% Interrupt Time",
          "% User Time",
          "% Processor Time"
        ],
        "resources": [
          "*"
        ],
        "append_dimensions": {
          "d1": "win_foo",
          "d2": "win_bar"
        }
      },
      "LogicalDisk": {
        "measurement": [
          {"name": "% Idle Time", "unit": "Percent"},
          {"name": "% Disk Read Time", "rename": "DISK_READ"},
          "% Disk Write Time"
        ],
        "resources": [
          "*"
        ]
      },
      "Memory": {
        "metrics_collection_interval": 5,
        "measurement": [
          "Available Bytes",
          "Cache Faults/sec",
```

```
          "Page Faults/sec",
          "Pages/sec"
        ],
        "append_dimensions": {
          "d3": "win_bo"
        }
      },
      "Network Interface": {
        "metrics_collection_interval": 5,
        "measurement": [
          "Bytes Received/sec",
          "Bytes Sent/sec",
          "Packets Received/sec",
          "Packets Sent/sec"
        ],
        "resources": [
          "*"
        ],
        "append_dimensions": {
          "d3": "win_bo"
        }
      },
      "System": {
        "measurement": [
          "Context Switches/sec",
          "System Calls/sec",
          "Processor Queue Length"
        ],
        "append_dimensions": {
          "d1": "win_foo",
          "d2": "win_bar"
        }
      }
    },
    "append_dimensions": {
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}",
      "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
    },
    "aggregation_dimensions" : [["ImageId"], ["InstanceId", "InstanceType"], ["d1"],[]]
    }
  }
```

CloudWatch Agenten-Konfigurationsdatei: Abschnitt „Protokolle“

Der Abschnitt `logs` enthält die folgenden Felder:

- `logs_collected` – Erforderlich, sofern der Abschnitt `logs` enthalten ist. Gibt an, welche Protokolldateien und Windows-Ereignisprotokolle vom Server erfasst werden sollen. Kann zwei Felder enthalten: `files` und `windows_events`.
  - `files`— Gibt an, welche regulären Protokolldateien der CloudWatch Agent sammeln soll. Enthält das Feld `collect_list`, das diese Dateien genauer definiert.
    - `collect_list` – Erforderlich, wenn `files` enthalten ist. Enthält ein Array von Einträgen, die jeweils eine zu erfassende Protokolldatei angeben. Jeder dieser Einträge kann die folgenden Felder enthalten:
      - `file_path`— Gibt den Pfad der Protokolldatei an, die in CloudWatch Logs hochgeladen werden soll. Globale standardmäßige Unixabgleichsregeln werden akzeptiert. `**` muss als super asterisk hinzugefügt werden. Beispiel: Angeben von `/var/log/**.log` bewirkt, dass alle `.log`-Dateien in der `/var/log`-Verzeichnisstruktur erfasst werden. Weitere Beispiele finden Sie in der globalen Bibliothek.

        Sie können das Standardsternchen auch als Standardplatzhalter verwenden. Beispiel: `/var/log/system.log*` findet Dateien wie `system.log_1111`, `system.log_2222` usw. in `/var/log`.

        Nur die neueste Datei wird basierend auf der Änderungszeit der Datei in die CloudWatch Logs übertragen. Wir empfehlen, dass Sie eine Reihe von Platzhaltern angeben, z. B. Dateien desselben Typs, `access_log.2018-06-01-01` und `access_log.2018-06-01-02`, aber nicht mehrere Arten von Dateien, wie z. B. `access_log_80` und `access_log_443`. Wenn Sie mehrere Arten von Dateien angeben möchten, fügen Sie der Agentenkonfigurationsdatei einen anderen Protokoll-Stream-Eintrag hinzu, damit jede Art von Protokolldatei in einen anderen Protokoll-Stream gestellt wird.
      - `auto_removal` – Optional. Wenn dies der Fall ist`true`, löscht der CloudWatch Agent diese Protokolldatei nach dem Lesen automatisch und sie wurde rotiert. Normalerweise werden die Protokolldateien gelöscht, nachdem ihr gesamter Inhalt in CloudWatch Logs hochgeladen wurde. Wenn der Agent jedoch das EOF (Dateiende) erreicht und auch eine andere neuere Protokolldatei entdeckt, die dieser entsprichtfile_path, löscht der Agent die ALTE Datei. Sie müssen also sicherstellen, dass Sie mit dem Schreiben in die ALTE Datei fertig sind, bevor Sie die NEUE Datei erstellen. Die RUST-Tracing-Bibliothek weist

eine bekannte Inkompatibilität auf, da sie möglicherweise eine NEUE Protokolldatei erstellt und dann trotzdem versucht, in die ALTE Protokolldatei zu schreiben.

Der Agent entfernt aus Protokollen, die mehrere Dateien erstellen, nur vollständige Dateien. Dies sind z. B. Protokolle, die für jedes Datum separate Dateien erstellen. Wenn ein Protokoll kontinuierlich in eine einzige Datei schreibt, wird es nicht entfernt.

Wenn Sie bereits eine Rotations- oder Entfernungsmethode für Protokolldateien eingerichtet haben, sollten Sie dieses Feld auslassen oder auf `false` festlegen.

Wenn Sie dieses Feld auslassen, wird der Standardwert `false` verwendet.

- `log_group_name` – Optional. Gibt an, was als Loggruppenname in CloudWatch Logs verwendet werden soll.

  Wir empfehlen, in diesem Feld einen Protokollgruppen-Namen festzulegen, um Verwirrungen zu vermeiden. Wenn Sie `log_group_name` auslassen, wird der Wert von `file_path` bis zu dem letzten Punkt als Name der Protokollgruppe verwendet. Beispiel: Falls der Dateipfad `/tmp/TestLogFile.log.2017-07-11-14` ist, lautet der Name der Protokollgruppe `/tmp/TestLogFile.log`.

  Wenn Sie einen Protokollgruppen-Namen angeben, können Sie `{instance_id}`, `{hostname}`, `{local_hostname}` und `{ip_address}` als Variablen im Namen verwenden. `{hostname}` ruft den Hostnamen aus den EC2-Metadaten ab und `{local_hostname}` verwendet den Hostnamen aus der Netzwerkkonfigurationsdatei.

  Wenn Sie diese Variablen verwenden, um viele verschiedene Loggruppen zu erstellen, beachten Sie die Begrenzung auf 1.000.000 Loggruppen pro Region und Konto.

  Zulässige Zeichen sind a – z, A – Z, 0 – 9, „_“ (Unterstrich), „-“ (Bindestrich), „/“ (Schrägstrich) und „.“ (Punkt).

- `log_group_class` Optional. Gibt an, welche Protokollgruppen-Klasse für die neue Protokollgruppe verwendet werden soll. Weitere Hinweise zu Protokollgruppen-Klassen finden Sie unter Protokollklassen.

  Gültige Werte sind `STANDARD` und `INFREQUENT_ACCESS`. Wenn Sie dieses Feld auslassen, wird der Standard `STANDARD` verwendet.

> **Important**
>
> Nachdem eine Protokollgruppe erstellt wurde, kann ihre Klasse nicht mehr geändert werden.

- `log_stream_name` Optional. Gibt an, was als Name des Protokolldatenstroms in CloudWatch Logs verwendet werden soll. Sie können die Variablen `{instance_id}`, `{hostname}`, `{local_hostname}` und `{ip_address}` im Namen verwenden. `{hostname}` ruft den Hostnamen aus den EC2-Metadaten ab, `{local_hostname}` verwendet den Hostnamen aus der Netzwerkkonfigurationsdatei.

  Wenn Sie dieses Feld auslassen, wird der Wert des Parameters `log_stream_name` im globalen `logs`-Abschnitt verwendet. Wird dies ebenfalls weggelassen, wird der Standardwert von `{instance_id}` verwendet.

  Wenn ein Protokoll-Stream noch nicht vorhanden ist, wird er automatisch erstellt.

- `retention_in_days` – Optional. Gibt an, wie viele Tage lang die Protokollereignisse in der festgelegten Protokollgruppe aufbewahrt werden.
  - Wenn der Agent diese Protokollgruppe jetzt erstellt und Sie dieses Feld auslassen, wird die Aufbewahrung dieser neuen Protokollgruppe auf niemals ablaufend gesetzt.
  - Wenn diese Protokollgruppe bereits vorhanden ist und Sie dieses Feld angeben, wird die von Ihnen angegebene neue Aufbewahrung verwendet. Wenn Sie dieses Feld für eine bereits vorhandene Protokollgruppe weglassen, wird die Aufbewahrung der Protokollgruppe nicht geändert.

    Der CloudWatch Agent-Assistent verwendet `-1` als Standardwert für dieses Feld, wenn es zur Erstellung der Agent-Konfigurationsdatei verwendet wird und Sie keinen Wert für die Protokollspeicherung angeben. Dieser vom Assistenten festgelegte `-1` Wert gibt an, dass die Ereignisse in der Protokollgruppe niemals ablaufen. Das manuelle Ändern dieses Werts auf `-1` hat jedoch keine Auswirkung.

  Mögliche Werte sind 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 2192, 2557, 2922, 3288 und 3653.

  Wenn Sie den Agenten so konfigurieren, dass er mehrere Protokollstreams in dieselbe Protokollgruppe schreibt, und Sie `retention_in_days` an einer Stelle angeben, wird damit die Protokollaufbewahrung für die gesamte Protokollgruppe festgelegt. Wenn Sie

`retention_in_days` an mehreren Stellen für dieselbe Protokollgruppe angeben und alle diese Werte gleich sind, wird die Aufbewahrung festgelegt. Werden jedoch für dieselbe Protokollgruppe an mehreren Stellen unterschiedliche `retention_in_days`-Werte angegeben, wird die Protokollaufbewahrung nicht festgelegt und der Agent wird gestoppt und gibt einen Fehler zurück.

> **Note**
>
> Die IAM-Rolle oder der IAM-Benutzer des Agenten muss über `logs:PutRetentionPolicy` verfügen, damit sie bzw. er Aufbewahrungsrichtlinien festlegen kann. Weitere Informationen finden Sie unter Erlauben Sie dem CloudWatch Agenten, eine Richtlinie zur Aufbewahrung von Protokollen festzulegen.

> **Warning**
>
> Wenn Sie für eine bereits vorhandene Protokollgruppe `retention_in_days` festlegen, werden alle Protokolle in dieser Protokollgruppe, die vor der von Ihnen angegebenen Anzahl von Tagen veröffentlicht wurden, gelöscht. Wenn Sie beispielsweise den Wert 3 festlegen, werden alle Protokolle gelöscht, die 3 oder mehr Tage alt sind.

- `filters` Optional. Kann ein Array von Einträgen enthalten, von denen jeder einen regulären Ausdruck und einen Filtertyp angibt, um anzugeben, ob Protokolleinträge, die dem Filter entsprechen, veröffentlicht oder gelöscht werden sollen. Wenn Sie dieses Feld weglassen, werden alle Protokolle in der Protokolldatei in CloudWatch Logs veröffentlicht. Wenn Sie dieses Feld angeben, verarbeitet der Agent jede Protokollnachricht mit allen von Ihnen angegebenen Filtern, und nur die Protokollereignisse, die alle Filter bestehen, werden in CloudWatch Logs veröffentlicht. Die Protokolleinträge, die nicht alle Filter bestehen, verbleiben weiterhin in der Protokolldatei des Hosts, werden aber nicht an CloudWatch Logs gesendet.

  Jeder Eintrag im Filter-Array kann die folgenden Felder enthalten:

  - `type` – Gibt den Filtertyp an. Gültige Werte sind `include` und `exclude`. Bei `include` muss der Protokolleintrag mit dem Ausdruck übereinstimmen, der in CloudWatch Logs veröffentlicht werden soll. Bei `exclude` wird nicht jeder Protokolleintrag, der dem Filter entspricht, an CloudWatch Logs gesendet.

- `expression` – Eine Zeichenfolge mit einem regulären Ausdruck, die der RE2-Syntax folgt.

> **Note**
>
> Der CloudWatch Agent überprüft nicht die Leistung eines von Ihnen angegebenen regulären Ausdrucks und schränkt auch nicht die Laufzeit der Auswertung der regulären Ausdrücke ein. Wir empfehlen, darauf zu achten, keinen Ausdruck zu erstellen, dessen Auswertung hohe Kosten verursacht. Weitere Informationen zu möglichen Problemen finden Sie unter Denial of Service für reguläre Ausdrücke — ReDo S

Der folgende Auszug aus der CloudWatch Agentenkonfigurationsdatei veröffentlicht beispielsweise Protokolle, bei denen es sich um PUT- und POST-Anfragen handelt, in CloudWatch Logs, jedoch keine Protokolle, die von Firefox stammen.

```
"collect_list": [
  {
    "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/test.log",
    "log_group_name": "test.log",
    "log_stream_name": "test.log",
    "filters": [
      {
        "type": "exclude",
        "expression": "Firefox"
      },
      {
        "type": "include",
        "expression": "P(UT|OST)"
      }
    ]
  },
  .....
]
```

> **Note**
>
> Die Reihenfolge der Filter in der Konfigurationsdatei ist für die Leistung wichtig. Im vorherigen Beispiel löscht der Agent alle Protokolle, die mit `Firefox`

übereinstimmen, bevor er mit der Auswertung des zweiten Filters beginnt. Wenn weniger Protokolleinträge mit mehreren Filtern ausgewertet werden sollen, platzieren Sie den Filter, der voraussichtlich mehr Protokolle ausschließt, an der ersten Stelle in der Konfigurationsdatei.

- `timezone` – Optional. Gibt die Zeitzone an, die verwendet werden soll, wenn Zeitstempel auf Protokollereignisse angewendet werden. Die gültigen Werte sind `UTC` und `Local`. Der Standardwert ist `Local`.

  Dieser Parameter wird ignoriert, wenn Sie keinen Wert für `timestamp_format` angeben.

- `timestamp_format` – Optional. Gibt das Zeitstempelformat an, wobei Klartext und spezielle Symbole verwendet werden, die mit % beginnen. Wenn Sie dieses Feld nicht ausfüllen, wird die aktuelle Zeit verwendet. Wenn Sie dieses Feld verwenden, können Sie die Symbole aus der folgenden Liste als Teil des Formats verwenden.

  Wenn ein einzelner Protokolleintrag zwei Zeitstempel enthält, die dem Format entsprechen, wird der erste Zeitstempel verwendet.

  Diese Liste von Symbolen unterscheidet sich von der Liste, die vom älteren CloudWatch Logs-Agenten verwendet wurde. Eine Zusammenfassung der Unterschiede finden Sie unter Zeitstempelunterschiede zwischen dem Unified CloudWatch Agent und dem früheren CloudWatch Logs-Agenten.

  `%y`

  Jahr ohne Jahrhundert als mit Nullen aufgefüllte Dezimalzahl Zum Beispiel, `19` um 2019 darzustellen.

  `%Y`

  Jahr mit Jahrhundert als Dezimalzahl z. B. `2019`.

  `%b`

  Monat als Abkürzung des Namens des Gebietsschemas

  `%B`

  Monat als vollständiger Name des Gebietsschemas

  `%m`

  Monat als mit Nullen aufgefüllte Dezimalzahl

`%-m`

Monat als Dezimalzahl (nicht mit Nullen aufgefüllt)

`%d`

Tag des Monats als mit Nullen aufgefüllte Dezimalzahl

`%-d`

Tag des Monats als Dezimalzahl (nicht mit Nullen aufgefüllt)

`%A`

Vollständiger Name des Wochentags, wie z. B. `Monday`

`%a`

Abkürzung des Wochentags, wie z. B. `Mon`

`%H`

Stunde (24-Stunden-Format) als mit Nullen aufgefüllte Dezimalzahl

`%I`

Stunde (12-Stunden-Format) als mit Nullen aufgefüllte Dezimalzahl

`%-I`

Stunde (12-Stunden-Format) als Dezimalzahl (nicht mit Nullen aufgefüllt)

`%p`

AM oder PM

`%M`

Minuten als mit Nullen aufgefüllte Dezimalzahl

`%-M`

Minuten als Dezimalzahl (nicht mit Nullen aufgefüllt)

`%S`

Sekunden als mit Nullen aufgefüllte Dezimalzahl

`%-S`

Sekunden als Dezimalzahl (nicht mit Nullen aufgefüllt)

`%f`

Sekundenbruchteile als Dezimalzahl (1 bis 9 Ziffern), links mit Nullen aufgefüllt.

`%Z`

Zeitzone, z. B. `PST`

`%z`

Zeitzone, ausgedrückt als der Abstand zwischen der lokalen Zeitzone und der Koordinierten Weltzeit (UTC). z. B. `-0700`. Es wird nur das dieses Format unterstützt. Beispiel: `-07:00` ist kein gültiges Format.

- `multi_line_start_pattern` – Gibt das Muster an, anhand dessen der Beginn einer Protokollmeldung identifiziert wird. Eine Protokollmeldung besteht aus einer Zeile, die mit dem angegebenen Muster übereinstimmt, und allen folgenden Zeilen, die nicht dem Muster entsprechen.

  Wenn Sie dieses Feld leer lassen, wird der Mehrzeilenmodus deaktiviert und bei jeder Zeile, die mit einem Zeichen beginnt, das kein Leerzeichen ist, wird der vorherige Protokolleintrag abgeschlossen und ein neuer Protokolleintrag gestartet.

  Wenn Sie dieses Feld aufnehmen, können Sie `{timestamp_format}` angeben, um den gleichen regulären Ausdruck wie Ihr Zeitstempelformat zu verwenden. Andernfalls können Sie für CloudWatch Logs einen anderen regulären Ausdruck angeben, anhand dessen die Startzeilen von mehrzeiligen Einträgen bestimmt werden.
- `encoding` – Gibt die Codierung der Protokolldatei an, damit sie korrekt gelesen werden kann. Wenn Sie eine falsche Codierung angeben, kann dies zu Datenverlust führen, weil Zeichen, die nicht decodiert werden können, durch andere Zeichen ersetzt werden.

  Der Standardwert ist `utf-8`. Die folgenden Werte sind möglich:

  `ascii, big5, euc-jp, euc-kr, gbk, gb18030, ibm866, iso2022-jp, iso8859-2, iso8859-3, iso8859-4, iso8859-5, iso8859-6, iso8859-7, iso8859-8, iso8859-8-i, iso8859-10, iso8859-13, iso8859-14, iso8859-15, iso8859-16, koi8-r, koi8-u, macintosh, shift_jis, utf-8,`

`utf-16, utf-16le, UTF-16, UTF-16LE, windows-874, windows-1250, windows-1251, windows-1252, windows-1253, windows-1254, windows-1255, windows-1256, windows-1257, windows-1258, x-mac-cyrillic`

- Im Abschnitt `windows_events` ist der Typ der Windows-Ereignisse angegeben, der von Servern mit Windows Server erfasst wird. Er enthält folgende Felder:
  - `collect_list` – Erforderlich, wenn `windows_events` enthalten ist. Gibt die Typen und Stufen von zu erfassenden Windows-Ereignissen an. Jedes zu erfassende Protokoll hat einen Eintrag in diesem Abschnitt, der folgende Felder enthalten kann:
    - `event_name` – Gibt den Typ der zu protokollierenden Windows-Ereignisse an. Dies entspricht dem Kanalnamen des Windows-Ereignisprotokolls, z. B. `System`, `Security`, `Application` usw. Dieses Feld ist für jeden Typ eines zu protokollierenden Windows-Ereignisses ein Pflichtfeld.

      > **Note**
      >
      > Wenn Nachrichten aus einem Windows-Protokollkanal CloudWatch abgerufen werden, wird der Protokollkanal anhand seiner `Full Name` Eigenschaft gesucht. Währenddessen zeigt der Navigationsbereich der Windows Event Viewer die `Log Name`-Eigenschaft von Protokollkanälen an. `Full Name` und `Log Name` stimmen nicht immer überein. Um den `Full Name` eines Kanals zu bestätigen, klicken Sie in der Windows Event Viewer mit der rechten Maustaste darauf und öffnen Sie Eigenschaften.

    - `event_levels` – Gibt die Ebenen des zu protokollierenden Ereignisses an. Sie müssen jede zu protokollierende Ebene angeben. Mögliche Werte sind `INFORMATION`, `WARNING`, `ERROR`, `CRITICAL` und `VERBOSE`. Dieses Feld ist für jeden Typ eines zu protokollierenden Windows-Ereignisses ein Pflichtfeld.
    - `log_group_name` – Erforderlich. Gibt an, was als Protokollgruppenname in CloudWatch Logs verwendet werden soll.
    - `log_stream_name` Optional. Gibt an, was als Name des Protokolldatenstroms in CloudWatch Logs verwendet werden soll. Sie können die Variablen `{instance_id}`, `{hostname}`, `{local_hostname}` und `{ip_address}` im Namen verwenden. `{hostname}` ruft den Hostnamen aus den EC2-Metadaten ab, `{local_hostname}` verwendet den Hostnamen aus der Netzwerkkonfigurationsdatei.

Wenn Sie dieses Feld auslassen, wird der Wert des Parameters `log_stream_name` im globalen `logs`-Abschnitt verwendet. Wird dies ebenfalls weggelassen, wird der Standardwert von `{instance_id}` verwendet.

Wenn ein Protokoll-Stream noch nicht vorhanden ist, wird er automatisch erstellt.

- `event_format` – Optional. Gibt das Format an, das beim Speichern von Windows-Ereignissen in CloudWatch Protokollen verwendet werden soll. `xml`verwendet das XML-Format wie in der Windows-Ereignisanzeige. `text`verwendet das CloudWatch Legacy-Logs-Agent-Format.
- `retention_in_days` Optional. Gibt an, wie viele Tage lang die Windows-Ereignisse in der festgelegten Protokollgruppe aufbewahrt werden.
  - Wenn der Agent diese Protokollgruppe jetzt erstellt und Sie dieses Feld auslassen, wird die Aufbewahrung dieser neuen Protokollgruppe auf niemals ablaufend gesetzt.
  - Wenn diese Protokollgruppe bereits vorhanden ist und Sie dieses Feld angeben, wird die von Ihnen angegebene neue Aufbewahrung verwendet. Wenn Sie dieses Feld für eine bereits vorhandene Protokollgruppe weglassen, wird die Aufbewahrung der Protokollgruppe nicht geändert.

    Der CloudWatch Agent-Assistent verwendet `-1` als Standardwert für dieses Feld, wenn es zur Erstellung der Agent-Konfigurationsdatei verwendet wird und Sie keinen Wert für die Aufbewahrung von Protokollen angeben. Dieser vom Assistenten festgelegte `-1`-Wert legt fest, dass die Ereignisse in der Protokollgruppe nicht ablaufen. Das manuelle Ändern dieses Werts auf `-1` hat jedoch keine Auswirkung.

  Mögliche Werte sind 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 2192, 2557, 2922, 3288 und 3653.

  Wenn Sie den Agenten so konfigurieren, dass er mehrere Protokollstreams in dieselbe Protokollgruppe schreibt, und Sie `retention_in_days` an einer Stelle angeben, wird damit die Protokollaufbewahrung für die gesamte Protokollgruppe festgelegt. Wenn Sie `retention_in_days` an mehreren Stellen für dieselbe Protokollgruppe angeben und alle diese Werte gleich sind, wird die Aufbewahrung festgelegt. Werden jedoch für dieselbe Protokollgruppe an mehreren Stellen unterschiedliche `retention_in_days`-Werte angegeben, wird die Protokollaufbewahrung nicht festgelegt und der Agent wird gestoppt und gibt einen Fehler zurück.

> **Note**
>
> Die IAM-Rolle oder der IAM-Benutzer des Agenten muss über `logs:PutRetentionPolicy` verfügen, damit sie bzw. er Aufbewahrungsrichtlinien festlegen kann. Weitere Informationen finden Sie unter Erlauben Sie dem CloudWatch Agenten, eine Richtlinie zur Aufbewahrung von Protokollen festzulegen.

> **Warning**
>
> Wenn Sie für eine bereits vorhandene Protokollgruppe `retention_in_days` festlegen, werden alle Protokolle in dieser Protokollgruppe, die vor der von Ihnen angegebenen Anzahl von Tagen veröffentlicht wurden, gelöscht. Wenn Sie beispielsweise den Wert 3 festlegen, werden alle Protokolle gelöscht, die 3 oder mehr Tage alt sind.

- `log_stream_name` – Erforderlich. Gibt den Namen des standardmäßigen Protokoll-Streams an, der für alle Protokolle oder Windows-Ereignisse verwendet werden soll, für die kein Name eines Protokoll-Streams im Parameter `log_stream_name` des zugehörigen Eintrags in `collect_list` definiert ist.
- `endpoint_override` – Gibt einen FIPS-Endpunkt oder einen privaten Link an, der als Endpunkt verwendet wird, an den der Agent Protokolle sendet. Wenn Sie dieses Feld angeben und einen privaten Link setzen, können Sie die Protokolle an einen Amazon-VPC-Endpunkt senden. Weitere Informationen finden Sie unter Was ist Amazon VPC?

  Der Wert von `endpoint_override` muss eine Zeichenkette sein, die eine URL ist.

  Beispielsweise legt der folgende Teil des Protokollabschnitts der Konfigurationsdatei fest, dass der Agent beim Senden von Protokollen einen VPC-Endpunkt verwendet.

```
{
  "logs": {
    "endpoint_override": "vpce-XXXXXXXXXXXXXXXXXXXXXXXX.logs.us-
east-1.vpce.amazonaws.com",
   ......
   },
}
```

- `force_flush_interval` – Gibt die maximale Zeitspanne in Sekunden an, in der Protokolle im Speicherpuffer verbleiben, bevor sie an den Server gesendet werden. Unabhängig von der Einstellung für dieses Feld werden die Protokolle an den Server gesendet, sobald die Größe der Protokolle im Puffer 1 MB erreicht. Der Standardwert ist 5.

  Wenn Sie den Agenten verwenden, um hochauflösende Metriken im eingebetteten Metrikformat zu melden, und Sie Alarme für diese Metriken einstellen, belassen Sie diesen Parameter auf dem Standardwert von 5. Andernfalls werden die Metriken mit einer Verzögerung gemeldet, die bei unvollständigen oder unvollständigen Daten zu einer Alarmierung führen kann.
- `credentials`— Gibt eine IAM-Rolle an, die beim Senden von Protokollen an ein anderes AWS Konto verwendet werden soll. Sofern angegeben, enthält dieses Feld einen Parameter, `role_arn`.
  - `role_arn`— Gibt den ARN einer IAM-Rolle an, der für die Authentifizierung verwendet werden soll, wenn Logs an ein anderes AWS Konto gesendet werden. Weitere Informationen finden Sie unter Metriken, Protokolle und Ablaufverfolgungen an ein anderes Konto senden. Wenn dieser Parameter hier angegeben wird, überschreibt er den Wert für `role_arn` im Abschnitt `agent` der Konfigurationsdatei, sofern vorhanden.
- `metrics_collected`— Dieses Feld kann Abschnitte enthalten, in denen angegeben wird, dass der Agent Protokolle sammeln soll, um Anwendungsfälle wie CloudWatch Application Signals und Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS zu ermöglichen.
  - `app_signals`(Optional) Gibt an, dass Sie CloudWatch Application Signals aktivieren möchten. Weitere Informationen zu dieser Konfiguration finden Sie unter CloudWatch Anwendungssignale aktivieren.
  - `kubernetes` – Dieses Feld kann einen `enhanced_container_insights`-Parameter enthalten, mit dem Sie Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS aktivieren können.
    - `enhanced_container_insights` – Stellen Sie dies auf `true` ein, um Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS zu aktivieren. Weitere Informationen finden Sie unter Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS.
    - `accelerated_compute_metrics`— Stellen Sie diese Option ein, `false` um die Erfassung von Nvidia-GPU-Metriken auf Amazon EKS-Clustern zu deaktivieren. Weitere Informationen finden Sie unter NVIDIA GPU-Metriken.
  - `emf` – Um in Protokollen eingebettete Metriken zu erfassen, ist es nicht mehr erforderlich, dieses `emf`-Feld hinzuzufügen. Dies ist ein Legacy-Feld, das angibt, dass der Agent Protokolle im eingebetteten Metrikformat erfassen soll. Sie können Metrikdaten aus diesen Protokollen generieren. Weitere Informationen finden Sie unter Einbetten von Metriken in Protokollen.

Es folgt ein Beispiel für den Abschnitt `logs`.

```
"logs":
   {
       "logs_collected": {
           "files": {
               "collect_list": [
                   {
                       "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
\Logs\\amazon-cloudwatch-agent.log",
                       "log_group_name": "amazon-cloudwatch-agent.log",
                       "log_stream_name": "my_log_stream_name_1",
                       "timestamp_format": "%H: %M: %S%y%b%-d"
                   },
                   {
                       "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
\Logs\\test.log",
                       "log_group_name": "test.log",
                       "log_stream_name": "my_log_stream_name_2"
                   }
               ]
           },
           "windows_events": {
               "collect_list": [
                   {
                       "event_name": "System",
                       "event_levels": [
                           "INFORMATION",
                           "ERROR"
                       ],
                       "log_group_name": "System",
                       "log_stream_name": "System"
                   },
                   {
                       "event_name": "CustomizedName",
                       "event_levels": [
                           "INFORMATION",
                           "ERROR"
                       ],
                       "log_group_name": "CustomizedLogGroup",
                       "log_stream_name": "CustomizedLogStream"
                   }
               ]
           }
```

```
        },
        "log_stream_name": "my_log_stream_name",
        "metrics_collected": {
            "kubernetes": {
                "enhanced_container_insights": true
       }
     }
   }
```

CloudWatch Agenten-Konfigurationsdatei: Abschnitt „Traces“

Indem Sie der CloudWatch Agentenkonfigurationsdatei einen `traces` Abschnitt hinzufügen, können Sie CloudWatch Application Signals aktivieren oder Traces von X-Ray und dem OpenTelemetry Instrumentierungs-SDK sammeln und an X-Ray senden.

> **Important**
>
> Die IAM-Rolle oder der IAM-Benutzer des Agenten müssen über die AWSXrayWriteOnlyAccessRichtlinie verfügen, um Trace-Daten an X-Ray zu senden. Weitere Informationen finden Sie unter Erstellen Sie IAM-Rollen und -Benutzer für die Verwendung mit dem Agenten CloudWatch .

Für einen schnellen Start mit der Erfassung von Traces können Sie der CloudWatch Agenten-Konfigurationsdatei einfach Folgendes hinzufügen.

```
"traces_collected": {
        "xray": {
        },
        "otlp": {
        }
      }
```

Wenn Sie den vorherigen Abschnitt zur CloudWatch Agentenkonfigurationsdatei hinzufügen und den Agenten neu starten, beginnt der Agent mit der Erfassung von Traces mit den folgenden Standardoptionen und -werten. Weitere Informationen zu diesen Parametern finden Sie in den Parameterdefinitionen weiter unten in diesem Abschnitt.

```
"traces_collected": {
        "xray": {
```

```
            "bind_address": "127.0.0.1:2000",
            "tcp_proxy": {
              "bind_address": "127.0.0.1:2000"
            }
        },
        "otlp": {
            "grpc_endpoint": "127.0.0.1:4317",
            "http_endpoint": "127.0.0.1:4318"
        }
      }
```

Der Abschnitt `traces` kann die folgenden Felder enthalten:

- `traces_collected` – Erforderlich, sofern der Abschnitt `traces` enthalten ist. Gibt an, von welchen SDKs Ablaufverfolgungen erfasst werden sollen. Dies kann die folgenden Felder umfassen:
  - `app_signals` Optional. Gibt an, dass Sie CloudWatch Anwendungssignale aktivieren möchten. Weitere Informationen zu dieser Konfiguration finden Sie unter CloudWatch Anwendungssignale aktivieren.
  - `xray` Optional. Gibt an, dass Sie Ablaufverfolgungen aus dem X-Ray-SDK sammeln möchten. Dieser Abschnitt kann die folgenden Felder enthalten:
    - `bind_address` – Optional. Gibt die UDP-Adresse an, die der CloudWatch Agent verwenden soll, um auf Röntgen-Traces zu warten. Das Format ist `ip:port`. Diese Adresse muss mit der im X-Ray SDK eingestellten Adresse übereinstimmen.

      Wenn Sie dieses Feld auslassen, wird der Standard `127.0.0.1:2000` verwendet.
    - `tcp_proxy` Optional. Konfiguriert die Adresse für einen Proxy, der für die Unterstützung von X-Ray Remote Sampling verwendet wird. Weitere Informationen finden Sie unter Konfigurieren von Samplingregeln in der X-Ray-Dokumentation.

      Dieser Abschnitt kann das folgende Feld enthalten.
      - `bind_address` Optional. Gibt die TCP-Adresse an, für die der CloudWatch Agent den Proxy einrichten soll. Das Format ist `ip:port`. Diese Adresse muss mit der im X-Ray SDK eingestellten Adresse übereinstimmen.

        Wenn Sie dieses Feld auslassen, wird der Standard `127.0.0.1:2000` verwendet.
  - `otlp` Optional. Gibt an, dass Sie Traces vom OpenTelemetry SDK sammeln möchten. Weitere Informationen zur Distribution für finden Sie unter AWS AWS Distro for OpenTelemetry.

OpenTelemetry Weitere Informationen zur AWS Distribution für OpenTelemetry SDKs finden Sie unter Einführung.

Dieser Abschnitt kann die folgenden Felder enthalten:

- `grpc_endpoint` – Optional. Gibt die Adresse an, die der CloudWatch Agent verwenden soll, um auf OpenTelemetry Traces zu warten, die mit gRPC Remote Procedure Calls gesendet wurden. Das Format ist `ip:port`. Diese Adresse muss mit der Adresse übereinstimmen, die für den gRPC-Exporter im OpenTelemetry SDK festgelegt wurde.

  Wenn Sie dieses Feld auslassen, wird der Standard `127.0.0.1:4317` verwendet.
- `http_endpoint` Optional. Gibt die Adresse an, die der CloudWatch Agent verwenden soll, um auf OTLP-Traces zu warten, die über HTTP gesendet wurden. Das Format ist `ip:port`. Diese Adresse muss mit der Adresse übereinstimmen, die für den HTTP-Exporter im OpenTelemetry SDK festgelegt wurde.

  Wenn Sie dieses Feld auslassen, wird der Standard `127.0.0.1:4318` verwendet.

- `concurrency` Optional. Gibt die maximale Anzahl der gleichzeitigen Aufrufe von X-Ray an, die zum Hochladen von Ablaufverfolgungen verwendet werden können. Der Standardwert ist 8
- `local_mode` Optional. Wenn `true`, erfasst der Agent keine Amazon-EC2-Instance-Metadaten. Der Standardwert ist `false`
- `endpoint_override` Optional. Gibt einen FIPS-Endpunkt oder einen privaten Link an, der als Endpunkt verwendet werden soll, an den der CloudWatch Agent Traces sendet. Wenn Sie dieses Feld angeben und einen privaten Link setzen, können Sie die Ablaufverfolgung an einen Amazon-VPC-Endpunkt senden. Weitere Informationen finden Sie unter Was ist Amazon VPC

  Der Wert von `endpoint_override` muss eine Zeichenkette sein, die eine URL ist.
- `region_override` Optional. Gibt die Region an, die für den X-Ray-Endpunkt verwendet werden soll. Der CloudWatch Agent sendet die Spuren in der angegebenen Region an X-Ray. Wenn Sie dieses Feld auslassen, sendet der Agent die Ablaufverfolgung an die Region, in der sich die Amazon-EC2-Instance befindet.

  Wenn Sie hier eine Region angeben, hat diese Vorrang vor der Einstellung des `region`-Parameters im `agent`-Abschnitt der Konfigurationsdatei.
- `proxy_override` Optional. Gibt die Proxy-Serveradresse an, die der CloudWatch Agent beim Senden von Anfragen an X-Ray verwenden soll. Das Protokoll des Proxyservers muss als Teil dieser Adresse angegeben werden.

- `credentials`— Gibt eine IAM-Rolle an, die beim Senden von Traces an ein anderes AWS Konto verwendet werden soll. Sofern angegeben, enthält dieses Feld einen Parameter, `role_arn`.
  - `role_arn`— Gibt den ARN einer IAM-Rolle an, der für die Authentifizierung verwendet werden soll, wenn Traces an ein anderes AWS Konto gesendet werden. Weitere Informationen finden Sie unter Metriken, Protokolle und Ablaufverfolgungen an ein anderes Konto senden. Wenn dieser Parameter hier angegeben wird, überschreibt er den Wert für `role_arn` im Abschnitt `agent` der Konfigurationsdatei, sofern vorhanden.

CloudWatch Agenten-Konfigurationsdatei: Vollständige Beispiele

Im Folgenden finden Sie ein Beispiel für eine vollständige CloudWatch Agentenkonfigurationsdatei für einen Linux-Server.

Die Elemente, die in den `measurement`-Abschnitten für die zu sammelnden Metriken aufgelistet werden, können entweder den vollständigen Namen angeben oder nur den Teil des Metriknamens, der an den Typ der Ressource angehängt wird. Beispiel: Die Angabe von entweder `reads` oder `diskio_reads` im Abschnitt `measurement` des Abschnitts `diskio` bewirkt, dass die Metrik `diskio_reads` erfasst wird.

Dieses Beispiel enthält beide Möglichkeiten zur Angabe von Metriken im Bereich `measurement`.

```
   {
     "agent": {
       "metrics_collection_interval": 10,
       "logfile": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log"
     },
     "metrics": {
       "namespace": "MyCustomNamespace",
       "metrics_collected": {
         "cpu": {
           "resources": [
             "*"
           ],
           "measurement": [
             {"name": "cpu_usage_idle", "rename": "CPU_USAGE_IDLE", "unit":
"Percent"},
             {"name": "cpu_usage_nice", "unit": "Percent"},
             "cpu_usage_guest"
           ],
           "totalcpu": false,
           "metrics_collection_interval": 10,
```

```
      "append_dimensions": {
        "customized_dimension_key_1": "customized_dimension_value_1",
        "customized_dimension_key_2": "customized_dimension_value_2"
      }
    },
    "disk": {
      "resources": [
        "/",
        "/tmp"
      ],
      "measurement": [
        {"name": "free", "rename": "DISK_FREE", "unit": "Gigabytes"},
        "total",
        "used"
      ],
       "ignore_file_system_types": [
        "sysfs", "devtmpfs"
      ],
      "metrics_collection_interval": 60,
      "append_dimensions": {
        "customized_dimension_key_3": "customized_dimension_value_3",
        "customized_dimension_key_4": "customized_dimension_value_4"
      }
    },
    "diskio": {
      "resources": [
        "*"
      ],
      "measurement": [
        "reads",
        "writes",
        "read_time",
        "write_time",
        "io_time"
      ],
      "metrics_collection_interval": 60
    },
    "swap": {
      "measurement": [
        "swap_used",
        "swap_free",
        "swap_used_percent"
      ]
    },
```

```
          "mem": {
            "measurement": [
              "mem_used",
              "mem_cached",
              "mem_total"
            ],
            "metrics_collection_interval": 1
          },
          "net": {
            "resources": [
              "eth0"
            ],
            "measurement": [
              "bytes_sent",
              "bytes_recv",
              "drop_in",
              "drop_out"
            ]
          },
          "netstat": {
            "measurement": [
              "tcp_established",
              "tcp_syn_sent",
              "tcp_close"
            ],
            "metrics_collection_interval": 60
          },
          "processes": {
            "measurement": [
              "running",
              "sleeping",
              "dead"
            ]
          }
        },
        "append_dimensions": {
          "ImageId": "${aws:ImageId}",
          "InstanceId": "${aws:InstanceId}",
          "InstanceType": "${aws:InstanceType}",
          "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
        },
        "aggregation_dimensions" : [["ImageId"], ["InstanceId", "InstanceType"],
["d1"],[]],
        "force_flush_interval" : 30
```

```
      },
      "logs": {
        "logs_collected": {
          "files": {
            "collect_list": [
              {
                "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-
agent.log",
                "log_group_name": "amazon-cloudwatch-agent.log",
                "log_stream_name": "amazon-cloudwatch-agent.log",
                "timezone": "UTC"
              },
              {
                "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/test.log",
                "log_group_name": "test.log",
                "log_stream_name": "test.log",
                "timezone": "Local"
              }
            ]
          }
        },
        "log_stream_name": "my_log_stream_name",
        "force_flush_interval" : 15,
        "metrics_collected": {
           "kubernetes": {
                "enhanced_container_insights": true
      }
    }
  }
}
```

Im Folgenden finden Sie ein Beispiel für eine vollständige CloudWatch Agentenkonfigurationsdatei für einen Server, auf dem Windows Server ausgeführt wird.

```
{
      "agent": {
        "metrics_collection_interval": 60,
        "logfile": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\\amazon-
cloudwatch-agent.log"
      },
      "metrics": {
        "namespace": "MyCustomNamespace",
        "metrics_collected": {
```

```
        "Processor": {
          "measurement": [
            {"name": "% Idle Time", "rename": "CPU_IDLE", "unit": "Percent"},
            "% Interrupt Time",
            "% User Time",
            "% Processor Time"
          ],
          "resources": [
            "*"
          ],
          "append_dimensions": {
            "customized_dimension_key_1": "customized_dimension_value_1",
            "customized_dimension_key_2": "customized_dimension_value_2"
          }
        },
        "LogicalDisk": {
          "measurement": [
            {"name": "% Idle Time", "unit": "Percent"},
            {"name": "% Disk Read Time", "rename": "DISK_READ"},
            "% Disk Write Time"
          ],
          "resources": [
            "*"
          ]
        },
        "customizedObjectName": {
          "metrics_collection_interval": 60,
          "customizedCounterName": [
            "metric1",
            "metric2"
          ],
          "resources": [
            "customizedInstances"
          ]
        },
        "Memory": {
          "metrics_collection_interval": 5,
          "measurement": [
            "Available Bytes",
            "Cache Faults/sec",
            "Page Faults/sec",
            "Pages/sec"
          ]
        },
```

```
          "Network Interface": {
            "metrics_collection_interval": 5,
            "measurement": [
              "Bytes Received/sec",
              "Bytes Sent/sec",
              "Packets Received/sec",
              "Packets Sent/sec"
            ],
            "resources": [
              "*"
            ],
            "append_dimensions": {
              "customized_dimension_key_3": "customized_dimension_value_3"
            }
          },
          "System": {
            "measurement": [
              "Context Switches/sec",
              "System Calls/sec",
              "Processor Queue Length"
            ]
          }
        },
        "append_dimensions": {
          "ImageId": "${aws:ImageId}",
          "InstanceId": "${aws:InstanceId}",
          "InstanceType": "${aws:InstanceType}",
          "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
        },
        "aggregation_dimensions" : [["ImageId"], ["InstanceId", "InstanceType"],
 ["d1"],[]]
      },
      "logs": {
        "logs_collected": {
          "files": {
            "collect_list": [
              {
                "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\
\amazon-cloudwatch-agent.log",
                "log_group_name": "amazon-cloudwatch-agent.log",
                "timezone": "UTC"
              },
              {
```

```
                "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\
\test.log",
                "log_group_name": "test.log",
                "timezone": "Local"
              }
            ]
          },
          "windows_events": {
            "collect_list": [
              {
                "event_name": "System",
                "event_levels": [
                  "INFORMATION",
                  "ERROR"
                ],
                "log_group_name": "System",
                "log_stream_name": "System",
                "event_format": "xml"
              },
              {
                "event_name": "CustomizedName",
                "event_levels": [
                  "WARNING",
                  "ERROR"
                ],
                "log_group_name": "CustomizedLogGroup",
                "log_stream_name": "CustomizedLogStream",
                "event_format": "xml"
              }
            ]
          }
        },
        "log_stream_name": "example_log_stream_name"
      }
    }
```

Speichern Sie die CloudWatch Agent-Konfigurationsdatei manuell

Wenn Sie die CloudWatch Agentenkonfigurationsdatei manuell erstellen oder bearbeiten, können Sie ihr einen beliebigen Namen geben. Zur Vereinfachung der Fehlerbehebung wird empfohlen, ihr auf Linux-Servern den Namen `/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json` und auf Servern, auf denen Windows Server ausgeführt wird, den Namen `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-`

`agent.json` zu geben. Anschließend können Sie die Datei auf andere Server kopieren, auf denen der Agent ausgeführt werden soll.

Die CloudWatch Agentenkonfigurationsdatei in den Systems Manager Parameter Store hochladen

Wenn Sie den SSM-Agent verwenden möchten, um den CloudWatch Agenten auf Servern zu installieren, können Sie die CloudWatch Agentenkonfigurationsdatei nach der manuellen Bearbeitung in den Systems Manager Parameter Store hochladen. Verwenden Sie dazu den Systems Manager `put-parameter`-Befehl.

Zum Speichern der Datei in Parameter Store müssen Sie eine IAM-Rolle mit ausreichenden Berechtigungen verwenden. Weitere Informationen finden Sie unter Erstellen Sie IAM-Rollen und -Benutzer für die Verwendung mit dem Agenten CloudWatch .

Verwenden Sie den folgenden Befehl, wobei *`parameter_name`* der für diese Datei zu verwendende Name in Parameter Store und *`configuration_file_pathname`* der Pfad und Dateiname der von Ihnen bearbeiteten Konfigurationsdatei ist.

```
aws ssm put-parameter --name "parameter name" --type "String" --value
 file://configuration_file_pathname
```

CloudWatch Anwendungssignale aktivieren

Verwenden Sie CloudWatch Application Signals, um Ihre Anwendungen automatisch zu nutzen, AWS sodass Sie die Anwendungsleistung anhand Ihrer Geschäftsziele verfolgen können. Application Signals bietet Ihnen eine einheitliche, anwendungsorientierte Ansicht Ihrer Java-Anwendungen, ihrer Abhängigkeiten und ihrer Edges. Weitere Informationen finden Sie unter Application Signals.

CloudWatch Application Signals nutzt den CloudWatch Agenten, um Metriken und Traces von Ihren automatisch instrumentierten Anwendungen zu empfangen, optional Regeln anzuwenden, um die hohe Kardinalität zu reduzieren, und dann die verarbeitete Telemetrie zu veröffentlichen. CloudWatch Mithilfe der Agenten-Konfigurationsdatei können Sie dem CloudWatch Agenten eine benutzerdefinierte Konfiguration speziell für Application Signals bereitstellen. Zunächst gibt das Vorhandensein eines `app_signals` Abschnitts unter dem Abschnitt innerhalb des `metrics_collected` `logs` Abschnitts der Agentenkonfigurationsdatei an, dass der CloudWatch Agent Metriken von Ihren automatisch instrumentierten Anwendungen empfängt. In ähnlicher Weise gibt das Vorhandensein eines `app_signals` Abschnitts unter dem `traces_collected` Abschnitt

innerhalb des `traces` Abschnitts der Agentenkonfigurationsdatei an, dass der CloudWatch Agent für den Empfang von Traces von Ihren automatisch instrumentierten Anwendungen aktiviert ist. Darüber hinaus können Sie optional benutzerdefinierte Konfigurationsregeln angeben, um die Veröffentlichung von Telemetriedaten mit hoher Kardinalität zu reduzieren, wie in diesem Abschnitt beschrieben.

- Wenn Sie das Amazon CloudWatch Observability EKS-Add-on für Amazon EKS-Cluster installieren, ist der CloudWatch Agent standardmäßig so aktiviert, dass er sowohl Metriken als auch Traces von Ihren automatisch instrumentierten Anwendungen empfängt. Wenn Sie optional benutzerdefinierte Konfigurationsregeln übergeben möchten, können Sie dies tun, indem Sie eine benutzerdefinierte Agentenkonfiguration an das Amazon-EKS-Add-On übergeben, wenn Sie es erstellen oder aktualisieren, indem Sie zusätzliche Konfigurationen verwenden, wie unter (Optional) Zusätzliche Konfiguration beschrieben.
- Für andere unterstützte Plattformen, einschließlich Amazon EC2, müssen Sie den CloudWatch Agenten mit einer Agentenkonfiguration starten, die Application Signals aktiviert, indem Sie die `app_signals` Abschnitte und optional alle benutzerdefinierten Konfigurationsregeln angeben, wie später in diesem Abschnitt beschrieben.

Im Folgenden finden Sie eine Übersicht über die Felder in der CloudWatch Agenten-Konfigurationsdatei, die sich auf CloudWatch Application Signals beziehen.

- `logs`
  - `metrics_collected`— Dieses Feld kann Abschnitte enthalten, in denen angegeben wird, dass der Agent Protokolle sammeln soll, um Anwendungsfälle wie CloudWatch Application Signals und Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS zu ermöglichen.

    > **Note**
    >
    > Bisher wurde dieser Abschnitt auch verwendet, um anzugeben, dass der Agent Protokolle im eingebetteten Metrikformat sammeln soll. Diese Einstellungen werden nicht mehr benötigt.

    - `app_signals`(Optional) Gibt an, dass Sie CloudWatch Application Signals ermöglichen möchten, Metriken von Ihren automatisch instrumentierten Anwendungen zu empfangen, um CloudWatch Application Signals zu unterstützen.

- `rules` (Optional) Eine Reihe von Regeln zur bedingten Auswahl von Metriken und Traces und zur Anwendung von Aktionen für Szenarien mit hoher Kardinalität. Jede Regel kann die folgenden Felder enthalten:
  - `rule_name` (Optional) Der Name der Regel.
  - `selectors` (Optional) Eine Reihe von Metriken und Traces mit Dimensionsabgleichungen. Jede Auswahl muss folgende Felder angeben:
    - `dimension` Erforderlich, wenn `selectors` nicht leer ist. Dies gibt die Dimension der Metriken und Traces an, die als Filter verwendet werden sollen.
    - `match` Erforderlich, wenn `selectors` nicht leer ist. Ein Platzhaltermuster, das zum Abgleichen von Werten der angegebenen Dimension verwendet wird.
  - `action` (Optional) Die Aktion, die auf Metriken und Traces angewendet werden soll, die den angegebenen Selektoren entsprechen. Der Wert von `action` muss eines der folgenden Schlüsselwörter sein:
    - `keep`Gibt an, dass nur die Metriken und Traces gesendet werden sollen, CloudWatch wenn sie mit den übereinstimmen. `selectors`
    - `drop` Gibt an, dass die Metrik und die Traces, die dem `selectors` entsprechen, verworfen werden sollen.
    - `replace` Gibt an, dass die Dimensionen der Metriken und Traces, die dem `selectors` entsprechen, ersetzt werden sollen. Sie werden entsprechend dem `replacements`-Abschnitt ersetzt.
  - `replacements`Erforderlich, wenn `action` ein `replace` ist. Eine Reihe von Dimensions- und Wertepaaren, die auf Metriken und Traces angewendet werden, die mit den angegebenen `selectors` übereinstimmen, wenn die `action` gleich `replace` ist. Jeder Ersatz muss folgende Felder angeben:
    - `target_dimension` Erforderlich, wenn `replacements` nicht leer ist. Gibt die Dimension an, die ersetzt werden muss.
    - `value` Erforderlich, wenn `replacements` nicht leer ist. Der Wert, durch den der ursprüngliche Wert von `target_dimension` ersetzt werden soll.
- `limiter`(Optional) Verwenden Sie diesen Abschnitt, um einzuschränken, an wie viele Metriken und Dimensionen Application Signals sendet CloudWatch, um Ihre Kosten zu optimieren.
  - `disabled`(Optional) Falls`true`, ist die Funktion zur Metrikbegrenzung deaktiviert. Der Standardwert ist `false`

- `drop_threshold`(Optional) Die maximale Anzahl unterschiedlicher Metriken pro Service in einem Rotationsintervall, die von einem CloudWatch Agenten exportiert werden können. Die Standardeinstellung ist 500.
- `rotation_interval`(Optional) Das Intervall, in dem der Limiter die Metrikdatensätze für die Differenzzählung zurücksetzt. Dies wird als Zeichenfolge mit einer Zahlenfolge und einem Einheitensuffix ausgedrückt. Brüche werden unterstützt. Die unterstützten Einheitensuffixe sinds,, mh, und `ms` `us` `ns`

  Die Standardeinstellung ist 1h für eine Stunde.
- `log_dropped_metrics`(Optional) Gibt an, ob der Agent Protokolle in die CloudWatch Agentenprotokolle schreiben soll, wenn Application Signals-Metriken gelöscht werden. Der Standardwert ist `false`.

> **Note**
>
> Um diese Protokollierung zu aktivieren, muss der `debug` Parameter im `agent` Abschnitt ebenfalls auf gesetzt sein`true`.

- `traces`
  - `traces_collected`
    - `app_signals` Optional. Geben Sie dies an, damit der CloudWatch Agent zur Unterstützung von CloudWatch Anwendungssignalen Traces von Ihren automatisch instrumentierten Anwendungen empfangen kann.

> **Note**
>
> Obwohl die benutzerdefinierten `app_signals`-Regeln in dem `metrics_collected`-Abschnitt angegeben sind, der im `logs`-Abschnitt enthalten ist, gelten sie auch implizit für den `traces_collected`-Abschnitt. Das gleiche Regelwerk gilt sowohl für Metriken als auch für Traces.

Wenn es mehrere Regeln mit unterschiedlichen Aktionen gibt, gelten sie in der folgenden Reihenfolge: `keep`, dann `drop`, dann `replace`.

Im Folgenden finden Sie ein Beispiel für eine vollständige CloudWatch Agentenkonfigurationsdatei, die benutzerdefinierte Regeln anwendet.

```
{
  "logs": {
    "metrics_collected": {
      "app_signals": {
        "rules": [
          {
            "rule_name": "keep01",
            "selectors": [
              {
                "dimension": "Service",
                "match": "pet-clinic-frontend"
              },
              {
                "dimension": "RemoteService",
                "match": "customers-service"
              }
            ],
            "action": "keep"
          },
          {
            "rule_name": "drop01",
            "selectors": [
              {
                "dimension": "Operation",
                "match": "GET /api/customer/owners/*"
              }
            ],
            "action": "drop"
          },
          {
            "rule_name": "replace01",
            "selectors": [
              {
                "dimension": "Operation",
                "match": "PUT /api/customer/owners/*/pets/*"
              },
              {
                "dimension": "RemoteOperation",
                "match": "PUT /owners"
              }
```

```
            ],
            "replacements": [
              {
                "target_dimension": "Operation",
                "value": "PUT /api/customer/owners/{ownerId}/pets{petId}"
              }
            ],
            "action": "replace"
          }
        ]
      }
    }
  },
  "traces": {
    "traces_collected": {
      "app_signals": {}
    }
  }
}
```

In der vorherigen Beispiel-Konfigurationsdatei werden die `rules` wie folgt verarbeitet:

1. Die Regel `keep01` stellt sicher, dass alle Metriken und Traces mit der Dimension `Service` als `pet-clinic-frontend` und der Dimension `RemoteService` als `customers-service` beibehalten werden.
2. Für die verarbeiteten Metriken und Traces nach der Anwendung von `keep01` stellt die `drop01`-Regel sicher, dass Metriken und Traces mit der Dimension `Operation` als `GET /api/customer/owners/*` verworfen werden.
3. Für die verarbeiteten Metriken und Traces nach der Anwendung von `drop01` aktualisiert die `replace01`-Regel Metriken und Traces, die die Dimension `Operation` als `PUT /api/customer/owners/*/pets/*` und die Dimension `RemoteOperation` als `PUT /owners` haben, sodass ihre `Operation`-Dimension jetzt durch `PUT /api/customer/owners/{ownerId}/pets{petId}` ersetzt wurde.

Im Folgenden finden Sie ein vollständiges Beispiel für eine CloudWatch Konfigurationsdatei, die die Kardinalität in Application Signals verwaltet, indem sie das Metriklimit auf 100 ändert, die Protokollierung gelöschter Metriken aktiviert und das Rotationsintervall auf zwei Stunden festlegt.

```
{
    "logs": {
```

```
        "metrics_collected": {
            "app_signals": {
                "limiter": {
                    "disabled": false,
                    "drop_threshold": 100,
                    "rotation_interval": "2h",
                    "log_dropped_metrics": true
                }
            }
        },
        "traces": {
            "traces_collected": {
                "app_signals": {}
            }
        }
    }
}
```

Netzwerkleistungsmetriken sammeln

EC2-Instances, die unter Linux ausgeführt werden, die den Elastic Network Adapter (ENA) verwenden, veröffentlichen Netzwerkleistungsmetriken. Version 1.246396.0 und höher des CloudWatch Agenten ermöglichen es Ihnen, diese Netzwerkleistungsmetriken in zu importieren. CloudWatch Wenn Sie diese Netzwerkleistungsmetriken in importieren CloudWatch, werden sie als benutzerdefinierte Messwerte berechnet. CloudWatch

Weitere Informationen zum ENA-Treiber finden Sie unter Erweiterte Netzwerke mit dem Elastic Network Adapter (ENA) auf Linux-Instances aktivieren und Erweiterte Netzwerke mit dem Elastic Network Adapter (ENA) auf Windows-Instances aktivieren.

Wie Sie die Sammlung von Netzwerkleistungsmetriken einrichten, unterscheidet sich von Linux-Servern und Windows-Servern.

In der folgenden Tabelle sind die vom ENA-Adapter aktivierten Netzwerkleistungsmetriken aufgeführt. Wenn der CloudWatch Agent diese Metriken CloudWatch aus Linux-Instances importiert, wird jeder dieser Metriknamen `ethtool_` am Anfang vorangestellt.

Metrik	Beschreibung
Name auf Linux-Servern: **`bw_in_allowance_exceeded`**	Die Anzahl der Pakete, die in die Warteschlange gestellt und/oder verworfen wurden, da die

Metrik	Beschreibung
Name auf Windows-Servern: **`Aggregate inbound BW allowance exceeded`**	eingehende aggregierte Bandbreite das Maximum für die Instance überschritten hat. Diese Metrik wird nur erfasst, wenn Sie sie im `ethtool` Unterabschnitt des `metrics_collected` Abschnitts der CloudWatch Agenten-Konfigurationsdatei aufgeführt haben. Weitere Informationen finden Sie unter Netzwerkleistungsmetriken sammeln. Einheit: keine
Name auf Linux-Servern: **`bw_out_allowance_exceeded`** Name auf Windows-Servern: **`Aggregate outbound BW allowance exceeded`**	Die Anzahl der Pakete, die in die Warteschlange gestellt und/oder verworfen wurden, weil die ausgehende aggregierte Bandbreite das Maximum für die Instance überschritten hat. Diese Metrik wird nur erfasst, wenn Sie sie im `ethtool` Unterabschnitt des `metrics_collected` Abschnitts der CloudWatch Agenten-Konfigurationsdatei aufgeführt haben. Weitere Informationen finden Sie unter Netzwerkleistungsmetriken sammeln. Einheit: keine

Metrik	Beschreibung
Name auf Linux-Servern: **`conntrack_allowance_available`** Name auf Windows-Servern: **`Available connection tracking allowance`**	Zeigt die Anzahl der nachverfolgten Verbindungen an, die von der Instance hergestellt werden können, bevor die zulässige Anzahl der nachverfolgten Verbindungen für diesen Instance-Typ erreicht wird. Diese Metrik ist nur auf Nitro-basierten EC2-Instances verfügbar, die den Linux-Treiber für Elastic Network Adapter (ENA) ab Version 2.8.1 verwenden, und auf Computern, die den Windows-Treiber für Elastic Network Adapter (ENA) ab Version 2.6.0 verwenden. Diese Metrik wird nur erfasst, wenn Sie sie im `ethtool` Unterabschnitt des Abschnitts der Agenten-Konfigurationsdatei `metrics_collected` aufgeführt haben. CloudWatch Weitere Informationen finden Sie unter Netzwerkleistungsmetriken sammeln. Einheit: keine

Metrik	Beschreibung
Name auf Linux-Servern: **`ena_srd_mode`** Name auf Windows-Servern: **`ena srd mode`**	Beschreibt, welche Funktionen von ENA Express aktiviert sind. Weitere Informationen zu ENA Express finden Sie unter Verbessern der Netzwerkleistung mit ENA Express auf Linux-Instances. Die Werte lauten wie folgt: • `0` = ENA Express aus, UDP aus • `1` = ENA Express ein, UDP aus • `2` = ENA Express aus, UDP ein **Note** Dies geschieht nur, wenn ENA Express ursprünglich aktiviert war und UDP so konfiguriert wurde, dass es verwendet wird. Der vorherige Wert wird für UDP-Verkehr beibehalten. • `3` = ENA Express ein, UDP ein

Metrik	Beschreibung
Name auf Linux-Servern: **ena_srd_eligible_tx_pkts** Name auf Windows-Servern: **ena srd eligible tx pkts**	Die Anzahl der innerhalb eines bestimmten Zeitraums gesendeten Netzwerkpakete, die die Zulassungsvoraussetzungen von AWS Scalable Reliable Datagram (SRD) erfüllen, wie folgt: • Sowohl sendende als auch empfangende Instance-Typen werden unterstützt. • Sowohl die sendenden als auch die empfangen den Instances müssen ENA Express konfiguriert haben. • Die sendenden und empfangenden Instances müssen sich im selben Subnetz befinden. • Der Netzwerkpfad zwischen den Instances darf keine Middleware-Boxen enthalten. ENA Express unterstützt derzeit keine Middleware-Boxen.
Name auf Linux-Servern: **ena_srd_tx_pkts** Name auf Windows-Servern: **ena srd tx pkts**	Die Anzahl der SRD-Pakete, die innerhalb eines bestimmten Zeitraums übertragen wurden.
Name auf Linux-Servern: **ena_srd_rx_pkts** Name auf Windows-Servern: **ena srd rx pkts**	Die Anzahl der innerhalb eines bestimmten Zeitraums empfangenen SRD-Pakete.
Name auf Linux-Servern: **ena_srd_resource_utilization** Name auf Windows-Servern: **ena srd resource utilization**	Der Prozentsatz der maximal zulässigen Speichera uslastung für gleichzeitige SRD-Verbindungen, den die Instance verbraucht hat.

Metrik	Beschreibung
Name auf Linux-Servern: **`linklocal_allowance_exceeded`** Name auf Windows-Servern: **`Link local packet rate allowance exceeded`**	Die Anzahl der verworfenen Pakete, weil das PPS des Datenverkehrs zu lokalen Proxy-Diensten das Maximum für die Netzwerkschnittstelle überschritten hat. Dies wirkt sich auf den Datenverkehr zum DNS-Dienst, zum Instance Metadata Service und zum Amazon Time Sync Service aus. Diese Metrik wird nur erfasst, wenn Sie sie im `ethtool` Unterabschnitt des `metrics_collected` Abschnitts der CloudWatch Agenten-Konfigurationsdatei aufgeführt haben. Weitere Informationen finden Sie unter Netzwerkleistungsmetriken sammeln. Einheit: keine
Name auf Linux-Servern: **`linklocal_allowance_exceeded`** Name auf Windows-Servern: **`Link local packet rate allowance exceeded`**	Die Anzahl der verworfenen Pakete, weil das PPS des Datenverkehrs zu lokalen Proxy-Diensten das Maximum für die Netzwerkschnittstelle überschritten hat. Dies wirkt sich auf den Datenverkehr zum DNS-Dienst, zum Instance Metadata Service und zum Amazon Time Sync Service aus. Diese Metrik wird nur erfasst, wenn Sie sie im `ethtool` Unterabschnitt des `metrics_collected` Abschnitts der CloudWatch Agenten-Konfigurationsdatei aufgeführt haben. Weitere Informationen finden Sie unter Netzwerkleistungsmetriken sammeln. Einheit: keine

Metrik	Beschreibung
Name auf Linux-Servern: **`pps_allowance_exceeded`** Name auf Windows-Servern: **`PPS allowance exceeded`**	Die Anzahl der Pakete, die in die Warteschlange gestellt und/oder verworfen wurden, weil die bidirektionale PPS das Maximum für die Instance überschritten hat. Diese Metrik wird nur erfasst, wenn Sie sie im `ethtool` Unterabschnitt des `metrics_collected` Abschnitts der CloudWatch Agenten-Konfigurationsdatei aufgeführt haben. Weitere Informationen finden Sie unter Netzwerkleistungsmetriken sammeln. Einheit: keine

Linux-Einrichtung

Auf Linux-Servern können Sie mit dem Ethtool-Plug-in die Netzwerkleistungsmetriken importieren. CloudWatch

ethtool ist ein Standard-Linux-Dienstprogramm, das Statistiken über Ethernet-Geräte auf Linux-Servern sammeln kann. Die erfassten Statistiken hängen vom Netzwerkgerät und vom Treiber ab. Beispiele für diese Statistiken sind`tx_cnt`, `rx_bytes`, `tx_errors` und `align_errors`. Wenn Sie das Ethtool-Plugin mit dem CloudWatch Agenten verwenden, können Sie diese Statistiken zusammen mit den weiter oben in CloudWatch diesem Abschnitt aufgeführten EC2-Netzwerkleistungskennzahlen auch in dieses importieren.

> **Tip**
>
> Verwenden Sie den Befehl `ethtool -S`, um die auf unserem Betriebssystem und Netzwerkgerät verfügbaren Statistiken zu finden.

Wenn der CloudWatch Agent Metriken importiert CloudWatch, fügt er den Namen aller importierten Metriken ein `ethtool_` Präfix hinzu. Also `rx_bytes` wird die standardmäßige Ethtool-Statistik aufgerufen `ethtool_rx_bytes` und die EC2-

Netzwerkleistungsmetrik `bw_in_allowance_exceeded` wird aufgerufen. CloudWatch `ethtool_bw_in_allowance_exceeded` CloudWatch

Um Ethtool-Metriken auf Linux-Servern zu importieren, fügen Sie dem `ethtool` Abschnitt der Agenten-Konfigurationsdatei einen `metrics_collected` Abschnitt hinzu. CloudWatch Der Abschnitt `ethtool` kann die folgenden Unterabschnitte enthalten:

- interface_include – Einschließen dieses Abschnitts bewirkt, dass der Agent Metriken nur von den Schnittstellen sammelt, deren Namen in diesem Abschnitt aufgeführt sind. Wenn Sie diesen Abschnitt auslassen, werden Metriken von allen Ethernet-Schnittstellen gesammelt, die nicht in `interface_exclude` aufgeführt sind.

  Die Standard-Ethernet-Schnittstelle ist `eth0`a.
- interface_exclude – Wenn Sie diesen Abschnitt einschließen, listen Sie die Ethernet-Schnittstellen auf, von denen Sie keine Metriken sammeln möchten.

  Das ethtool-Plug-In ignoriert immer Loopback-Schnittstellen.
- metrics_include — Dieser Abschnitt listet die Metriken auf, in die importiert werden soll. CloudWatch Es kann sowohl von ethtool gesammelte Standardstatistiken als auch hochauflösende Amazon-EC2-Netzwerkmetriken enthalten.

Im folgenden Beispiel wird ein Teil der Agenten-Konfigurationsdatei angezeigt. CloudWatch Diese Konfiguration erfasst die standardmäßigen Ethtool-Metriken `rx_packets` und `tx_packets` sowie die Amazon-EC2-Netzwerkleistungsmetriken nur von der `eth1`-Schnittstelle.

Weitere Informationen zur CloudWatch Agenten-Konfigurationsdatei finden Sie unter Erstellen oder bearbeiten Sie die CloudWatch Agenten-Konfigurationsdatei manuell.

```
"metrics": {
    "append_dimensions": {
      "InstanceId": "${aws:InstanceId}"
    },
    "metrics_collected": {
      "ethtool": {
        "interface_include": [
          "eth1"
        ],
        "metrics_include": [
          "rx_packets",
          "tx_packets",
```

```
            "bw_in_allowance_exceeded",
            "bw_out_allowance_exceeded",
            "conntrack_allowance_exceeded",
            "linklocal_allowance_exceeded",
            "pps_allowance_exceeded"
          ]
       }
   }
}
```

Windows-Einrichtung

Auf Windows-Servern sind die Netzwerkleistungsmesswerte über die Windows-Leistungsindikatoren verfügbar, von denen der CloudWatch Agent bereits Messwerte erfasst. Sie benötigen also kein Plugin, um diese Metriken von Windows-Servern zu sammeln.

Nachfolgend finden Sie eine Beispielkonfigurationsdatei zur Erfassung von Netzwerkleistungsmetriken von Windows. Weitere Informationen zum Bearbeiten der CloudWatch Agent-Konfigurationsdatei finden Sie unter Erstellen oder bearbeiten Sie die CloudWatch Agenten-Konfigurationsdatei manuell.

```
{
    "metrics": {
        "append_dimensions": {
            "InstanceId": "${aws:InstanceId}"
        },
        "metrics_collected": {
            "ENA Packets Shaping": {
                "measurement": [
                    "Aggregate inbound BW allowance exceeded",
                    "Aggregate outbound BW allowance exceeded",
                    "Connection tracking allowance exceeded",
                    "Link local packet rate allowance exceeded",
                    "PPS allowance exceeded"
                ],
                "metrics_collection_interval": 60,
                "resources": [
                    "*"
                ]
            }
        }
    }
```

```
}
```

Netzwerkleistungsmetriken anzeigen

Nachdem Sie die Netzwerkleistungsmetriken in importiert haben CloudWatch, können Sie sich diese Metriken als Zeitreihendiagramme ansehen und Alarme erstellen, die diese Metriken überwachen und Sie benachrichtigen, wenn sie einen von Ihnen festgelegten Schwellenwert überschreiten. Das folgende Verfahren zeigt, wie Sie ethtool-Metriken als Zeitreihendiagramm anzeigen. Weitere Informationen zum Einrichten eines -Alarms finden Sie unter CloudWatch Amazon-Alarme verwenden.

Da es sich bei all diesen Metriken um aggregierte Zähler handelt, können Sie mathematische Funktionen verwenden CloudWatch , `RATE(METRICS())` um z. B. die Rate dieser Metriken in Diagrammen zu berechnen oder sie zum Einstellen von Alarmen zu verwenden. Weitere Informationen zu Metrikberechnungsfunktionen finden Sie unter Verwenden von Metrikberechnungen

Um Netzwerkleistungsmetriken in der CloudWatch Konsole anzuzeigen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie den Namespace für die vom Agent zu erfassenden Metriken. Standardmäßig ist dies CWAgent, aber Sie haben möglicherweise einen anderen Namespace in der CloudWatch Agentenkonfigurationsdatei angegeben.
4. Wählen Sie eine Metrikdimension aus (z. B. Per-Instance Metrics (Metriken pro Instance)).
5. Die Registerkarte All metrics zeigt alle Metriken für diese Dimension im Namespace an. Sie haben die folgenden Möglichkeiten:

   a. Um eine Metrik grafisch darzustellen, müssen Sie das Kontrollkästchen neben der Metrik aktivieren. Um alle Metriken auszuwählen, aktivieren Sie das Kontrollkästchen in der Kopfzeile der Tabelle.

   b. Um die Tabelle sortieren, verwenden Sie die Spaltenüberschrift.

   c. Um nach Ressource zu filtern, müssen Sie zunächst die Ressourcen-ID und dann die Option Add to search (Zu Suche hinzufügen) wählen.

   d. Um nach Metrik zu filtern, müssen Sie den Metriknamen und anschließend Add to search (Zu Suche hinzufügen) wählen.
6. (Optional) Um dieses Diagramm zu einem CloudWatch Dashboard hinzuzufügen, wählen Sie Aktionen und dann Zum Dashboard hinzufügen aus.

Erfassen von NVIDIA GPU-Metriken

Sie können den CloudWatch Agenten verwenden, um NVIDIA-GPU-Metriken von Linux-Servern zu sammeln. Um dies einzurichten, fügen Sie dem `nvidia_gpu` `metrics_collected` Abschnitt der CloudWatch Agenten-Konfigurationsdatei einen Abschnitt hinzu. Weitere Informationen finden Sie unter Linux-Abschnitt.

Darüber hinaus muss auf der Instance ein NVIDIA-Treiber installiert sein. NVIDIA-Treiber sind auf einigen Amazon Machine Images (AMIs) vorinstalliert. Andernfalls können Sie den Treiber manuell installieren. Weitere Informationen finden Sie unter Installieren von NVIDIA-Treibern auf Linux-Instances.

Die folgenden Metriken können erfasst werden. Alle diese Metriken werden ohne Angabe erfasst CloudWatch `Unit`, aber Sie können für jede Metrik eine Einheit angeben, indem Sie der CloudWatch Agentenkonfigurationsdatei einen Parameter hinzufügen. Weitere Informationen finden Sie unter Linux-Abschnitt.

Metrik	Name der Metrik in CloudWatch	Beschreibung
`utilization_gpu`	`nvidia_smi_utilization_gpu`	Der Prozentsatz der Zeit im vergangenen Erfassungszeitraum, während dessen ein oder mehrere Kernel der GPU aktiv waren.
`temperature_gpu`	`nvidia_smi_temperature_gpu`	Die GPU-Kerntemperatur in Grad Celsius.
`power_draw`	`nvidia_smi_power_draw`	Die letzte gemessene Leistungsaufnahme des gesamten Boards in Watt.
`utilization_memory`	`nvidia_smi_utilization_memory`	Der Prozentsatz der Zeit im vergangenen Erfassungszeitraum, während dessen der globale Speicher (Gerätespeicher) gelesen oder geschrieben wurde.
`fan_speed`	`nvidia_smi_fan_speed`	Der Prozentsatz der maximalen Lüfterdrehzahl, mit der der Lüfter des Geräts derzeit laufen soll.

Metrik	Name der Metrik in CloudWatch	Beschreibung
memory_total	nvidia_smi_memory_total	Der gemeldete Gesamtspeicher in MB.
memory_used	nvidia_smi_memory_used	Der verwendete Speicher in MB.
memory_free	nvidia_smi_memory_free	Der freie Speicher in MB.
pcie_link_gen_current	nvidia_smi_pcie_link_gen_current	Die aktuelle Link-Generation.
pcie_link_width_current	nvidia_smi_pcie_link_width_current	Die aktuelle Link-Breite.
encoder_stats_session_count	nvidia_smi_encoder_stats_session_count	Aktuelle Anzahl von Encoder-Sitzungen.
encoder_stats_average_fps	nvidia_smi_encoder_stats_average_fps	Der gleitende Durchschnitt der Codierungs-Frames pro Sekunde.
encoder_stats_average_latency	nvidia_smi_encoder_stats_average_latency	Der gleitende Durchschnitt der Codier-Latenz in Mikrosekunden.
clocks_current_graphics	nvidia_smi_clocks_current_graphics	Die aktuelle Frequenz der Grafikuhr (Shader).

Metrik	Name der Metrik in CloudWatch	Beschreibung
clocks_current_sm	nvidia_smi_clocks_current_sm	Die aktuelle Frequenz der SM-Uhr (Streaming Multiprozessor).
clocks_current_memory	nvidia_smi_clocks_current_memory	Die aktuelle Frequenz der Speicheruhr.
clocks_current_video	nvidia_smi_clocks_current_video	Die aktuelle Frequenz der Videouhr (Encoder plus Decoder).

Alle diese Metriken werden mit den folgenden Dimensionen erfasst:

Dimension	Beschreibung
index	Ein eindeutiger Bezeichner für die GPU dieses Servers. Stellt den NVML-Index (NVIDIA Management Library) des Geräts dar.
name	Die Art der GPU. Zum Beispiel, NVIDIA Tesla A100
host	Der Hostname des Servers.

Erfassen von Prozessmetriken mit dem procstat-Plugin

Das procstat-Plugin ermöglicht es Ihnen, Metriken aus einzelnen Prozessen zu erfassen. Es wird auf Linux-Servern und auf Servern unterstützt, auf denen eine unterstützte Version von Windows Server ausgeführt wird.

Themen

Den CloudWatch Agenten für procstat konfigurieren

Um das procstat-Plug-In zu verwenden, fügen Sie dem `procstat` Abschnitt der CloudWatch Agenten-Konfigurationsdatei einen `metrics_collected` Abschnitt hinzu. Es gibt drei Möglichkeiten, die zu überwachenden Prozesse festzulegen. Sie können nur eine dieser Methoden verwenden, aber Sie können diese Methode verwenden, um einen oder mehrere Prozesse zur Überwachung anzugeben.

- `pid_file`: Wählt Prozesse anhand der Namen der von ihnen erstellten PID-Dateien (Process Identification Number) aus.
- `exe`: Wählt die Prozesse mit Prozessnamen aus, die mit der von Ihnen angegebenen Zeichenkette übereinstimmen, unter Verwendung von Abgleichsregeln für reguläre Ausdrücke. Die Übereinstimmung ist eine „enthält“ -Übereinstimmung, d. h., wenn Sie `agent` als übereinstimmender Begriff angeben, werden Prozesse mit Namen wie `cloudwatchagent` mit der Angabe übereinstimmen. Weitere Informationen finden Sie unter Syntax.
- `pattern`: Wählt Prozesse über die Befehlszeilen aus, die zum Starten der Prozesse verwendet werden. Es werden alle Prozesse ausgewählt, deren Befehlszeilen mit Hilfe von Regeln für den Abgleich mit regulären Ausdrücken mit der angegebenen Zeichenkette übereinstimmen. Die gesamte Befehlszeile wird überprüft, einschließlich der Parameter und Optionen, die mit dem Befehl verwendet werden.

  Die Übereinstimmung ist eine „enthält“ -Übereinstimmung, d. h., wenn Sie `-c` als abzugleichenden Ausdruck angeben, werden Prozesse mit Parametern wie `-config` als Übereinstimmung interpretiert.
- `drop_original_metrics` Optional. Wenn Sie das `aggregation_dimensions`-Feld im `metrics`-Abschnitt verwenden, um Metriken zu aggregierten Ergebnissen zusammenzufassen,

dann sendet der Agent standardmäßig sowohl die aggregierten Metriken als auch die ursprünglichen Metriken, die für jeden Wert der Dimension getrennt sind. Wenn Sie nicht möchten, dass die ursprünglichen Metriken gesendet werden CloudWatch, können Sie diesen Parameter mit einer Liste von Metriken angeben. Für die zusammen mit diesem Parameter angegebenen Metriken werden keine Kennzahlen nach Dimension gemeldet CloudWatch. Stattdessen werden nur die aggregierten Metriken gemeldet. Dadurch verringert sich die Anzahl der Metriken, die der Agent erfasst, was Ihre Kosten senkt.

Der CloudWatch Agent verwendet nur eine dieser Methoden, auch wenn Sie mehr als einen der obigen Abschnitte angeben. Wenn Sie mehr als einen Abschnitt angeben, verwendet der CloudWatch Agent den `pid_file` Abschnitt, sofern er vorhanden ist. Wenn nicht, verwendet er den `exe`-Abschnitt.

Auf Linux-Servern werden die Zeichenfolgen, die Sie in einem `exe`- oder `pattern`-Abschnitt angeben, als reguläre Ausdrücke ausgewertet. Auf Servern mit Windows Server werden diese Zeichenketten als WMI-Abfragen ausgewertet. Ein Beispiel wäre `pattern: "%apache%"`. Weitere Informationen finden Sie unter LIKE Operator (LIKE-Operator).

Welche Methode Sie auch immer verwenden, Sie können einen optionalen `metrics_collection_interval`-Parameter einschließen, der angibt, wie oft diese Metriken in Sekunden erfasst werden sollen. Wenn Sie diesen Parameter weglassen, wird der Standardwert von 60 Sekunden verwendet.

In den Beispielen in den folgenden Abschnitten ist der `procstat`-Abschnitt der einzige Abschnitt, der im `metrics_collected`-Abschnitt der Agent-Konfigurationsdatei enthalten ist. Tatsächliche Konfigurationsdateien können auch andere Abschnitte in `metrics_collected` enthalten. Weitere Informationen finden Sie unter Erstellen oder bearbeiten Sie die CloudWatch Agenten-Konfigurationsdatei manuell.

Konfigurieren mit pid_file

Das folgende Beispiel eines `procstat`-Abschnitts überwacht die Prozesse, die die PID-Dateien `example1.pid` und `example2.pid` erstellen. Von jedem Prozess werden unterschiedliche Metriken erfasst. Metriken, die aus dem Prozess erfasst wurden, der `example2.pid` erstellt, werden alle 10 Sekunden erfasst, während die Metriken aus dem Prozess `example1.pid` alle 60 Sekunden (Standardwert) erfasst werden.

```
{
    "metrics": {
```

```
        "metrics_collected": {
            "procstat": [
                {
                    "pid_file": "/var/run/example1.pid",
                    "measurement": [
                        "cpu_usage",
                        "memory_rss"
                    ]
                },
                {
                    "pid_file": "/var/run/example2.pid",
                    "measurement": [
                        "read_bytes",
                        "read_count",
                        "write_bytes"
                    ],
                    "metrics_collection_interval": 10
                }
            ]
        }
    }
}
```

Konfigurieren mit exe

Das folgende Beispiel eines `procstat`-Abschnitts überwacht alle Prozesse mit Namen, die mit den Zeichenfolgen `agent` oder `plugin` übereinstimmen. Von jedem Prozess werden die gleichen Metriken gesammelt.

```
{
    "metrics": {
        "metrics_collected": {
            "procstat": [
                {
                    "exe": "agent",
                    "measurement": [
                        "cpu_time",
                        "cpu_time_system",
                        "cpu_time_user"
                    ]
                },
                {
                    "exe": "plugin",
```

```
                    "measurement": [
                        "cpu_time",
                        "cpu_time_system",
                        "cpu_time_user"
                    ]
                }
            ]
        }
    }
}
```

Konfigurieren mit Muster

Das folgende Beispiel eines `procstat`-Abschnitts überwacht alle Prozesse mit Befehlszeilen, die den Zeichenfolgen `config` oder `-c` entsprechen. Von jedem Prozess werden die gleichen Metriken gesammelt.

```
{
    "metrics": {
        "metrics_collected": {
            "procstat": [
                {
                    "pattern": "config",
                    "measurement": [
                        "rlimit_memory_data_hard",
                        "rlimit_memory_data_soft",
                        "rlimit_memory_stack_hard",
                        "rlimit_memory_stack_soft"
                    ]
                },
                {
                    "pattern": "-c",
                    "measurement": [
                        "rlimit_memory_data_hard",
                        "rlimit_memory_data_soft",
                        "rlimit_memory_stack_hard",
                        "rlimit_memory_stack_soft"
                    ]
                }
            ]
        }
    }
}
```

Von Procstat erfasste Metriken

Die folgende Tabelle listet die Metriken auf, die Sie mit dem `procstat`-Plug-in erfassen können.

Der CloudWatch Agent fügt `procstat` am Anfang der folgenden Metriknamen hinzu. Es gibt eine unterschiedliche Syntax, je nachdem, ob sie von einem Linux-Server oder einem Server mit Windows-Server erfasst wurde. Zum Beispiel erscheint die `cpu_time`-Metrik als `procstat_cpu_time`, wenn sie unter Linux erfasst wurde, aber als `procstat cpu_time`, wenn sie unter Windows Server erfasst wurde.

Metrikname	Verfügbar am	Beschreibung
`cpu_time`	Linux	Die Zeit, die der Prozess die CPU nutzt. Diese Metrik wird in Hundertstelsekunden gemessen. Einheit: Anzahl
`cpu_time_guest`	Linux	Die Zeitspanne, die der Vorgang im Gastmodus ist. Diese Metrik wird in Hundertstelsekunden gemessen. Typ: Float Einheit: keine
`cpu_time_guest_nice`	Linux	Die Zeitspanne, die der

Metrikname	Verfügbar am	Beschreibung
		Prozess in einem Niced Guest läuft. Diese Metrik wird in Hundertstelsekunden gemessen. Typ: Float Einheit: keine
`cpu_time_idle`	Linux	Die Zeitspanne, die der Prozess im Ruhezustand ist. Diese Metrik wird in Hundertstelsekunden gemessen. Typ: Float Einheit: keine

Metrikname	Verfügbar am	Beschreibung
`cpu_time_iowait`	Linux	Die Zeitspanne, die der Prozess auf den Abschluss von E/A-Operationen wartet. Diese Metrik wird in Hundertstelsekunden gemessen. Typ: Float Einheit: keine
`cpu_time_irq`	Linux	Die Zeitspanne, in der der Prozess Unterbrechungen behandelt. Diese Metrik wird in Hundertstelsekunden gemessen. Typ: Float Einheit: keine

Metrikname	Verfügbar am	Beschreibung
cpu_time_nice	Linux	Die Zeitspanne, in der sich der Prozess im Nice Mode befindet. Diese Metrik wird in Hundertstelsekunden gemessen. Typ: Float Einheit: keine
cpu_time_soft_irq	Linux	Die Zeitspanne, in der der Prozess Software-Unterbrechungen behandelt. Diese Metrik wird in Hundertstelsekunden gemessen. Typ: Float Einheit: keine

Metrikname	Verfügbar am	Beschreibung
`cpu_time_steal`	Linux	Die Zeitspanne, die in anderen Betriebssystemen verbracht wird, wenn das Programm in einer virtualisierten Umgebung läuft. Diese Metrik wird in Hundertstelsekunden gemessen. Typ: Float Einheit: keine

Metrikname	Verfügbar am	Beschreibung
`cpu_time_stolen`	Linux, Windows Server	Die Zeitspanne, in der sich der Prozess in gestohlener Zeit befindet, also die Zeit, die in anderen Betriebssystemen in einer virtualisierten Umgebung verbracht wird. Diese Metrik wird in Hundertstelsekunden gemessen. Typ: Float Einheit: keine
`cpu_time_system`	Linux, Windows Server, macOS	Die Zeit, die sich der Prozess im Systemmodus befindet. Diese Metrik wird in Hundertstelsekunden gemessen. Typ: Float Einheit: Anzahl

Metrikname	Verfügbar am	Beschreibung
`cpu_time_user`	Linux, Windows Server, macOS	Die Zeit, die sich der Prozess im Benutzermodus befindet. Diese Metrik wird in Hundertstelsekunden gemessen. Einheit: Anzahl
`cpu_usage`	Linux, Windows Server, macOS	Der Prozentsatz der Zeit, in der der Prozess in beliebiger Funktion aktiv ist. Einheit: Prozent
`memory_data`	Linux, macOS	Die Menge an Speicher, die der Prozess für Daten benötigt. Einheit: Byte

Metrikname	Verfügbar am	Beschreibung
`memory_locked`	Linux, macOS	Die Menge an Speicher, die der Prozess gesperrt hat. Einheit: Byte
`memory_rss`	Linux, Windows Server, macOS	Die Menge des realen Speichers (resident set), die der Prozess verwendet. Einheit: Byte
`memory_stack`	Linux, macOS	Die Menge an Stackspeicher, die der Prozess verwendet. Einheit: Byte
`memory_swap`	Linux, macOS	Die Menge an Auslagerungsspeicher, die der Prozess verwendet. Einheit: Byte

Metrikname	Verfügbar am	Beschreibung
`memory_vms`	Linux, Windows Server, macOS	Die Menge an virtuellem Speicher, die der Prozess verwendet. Einheit: Byte
`num_fds`	Linux	Die Anzahl der Dateideskriptoren, die dieser Prozess geöffnet hat. Einheit: keine
`num_threads`	Linux, Windows, MacOS	Die Anzahl der Threads in diesem Prozess. Einheit: keine
`pid`	Linux, Windows Server, macOS	Prozesskennung (ID). Einheit: keine

Metrikname	Verfügbar am	Beschreibung
`pid_count`	Linux, Windows Server, macOS	Die Anzahl der dem Prozess zugeordneten Prozess-IDs. Auf Linux-Servern und macOS-Computern lautet der volle Name dieser Metrik `procstat_lookup_pid_count` und auf Windows-Servern `procstat_lookup pid_count` . Einheit: keine
`read_bytes`	Linux, Windows Server	Die Anzahl der Bytes, die der Prozess von Datenträgern gelesen hat. Einheit: Byte

Metrikname	Verfügbar am	Beschreibung
`write_bytes`	Linux, Windows Server	Die Anzahl der Bytes, die der Prozess auf Datenträger geschrieben hat. Einheit: Byte
`read_count`	Linux, Windows Server	Die Anzahl der Datenträg erlesevor gänge, die der Prozess ausgeführt hat. Einheit: keine
`rlimit_realtime_pr iority_hard`	Linux	Das feste Limit für die Echtzeitp riorität, die für diesen Prozess festgelegt werden kann. Einheit: keine

Metrikname	Verfügbar am	Beschreibung
`rlimit_realtime_priority_soft`	Linux	Das weiche Limit für die Echtzeitpriorität, die für diesen Prozess festgelegt werden kann. Einheit: keine
`rlimit_signals_pending_hard`	Linux	Das feste Limit für die maximale Anzahl von Signalen, die von diesem Prozess in die Warteschlange gestellt werden können. Einheit: keine
`rlimit_signals_pending_soft`	Linux	Das weiche Limit für die maximale Anzahl von Signalen, die von diesem Prozess in die Warteschlange gestellt werden können. Einheit: keine

Metrikname	Verfügbar am	Beschreibung
`rlimit_nice_priority_hard`	Linux	Das feste Limit für die maximale Nice-Priorität, die von diesem Prozess festgelegt werden kann. Einheit: keine
`rlimit_nice_priority_soft`	Linux	Das weiche Limit für die maximale Nice-Priorität, die von diesem Prozess festgelegt werden kann. Einheit: keine
`rlimit_num_fds_hard`	Linux	Das feste Limit für die maximale Anzahl von Dateideskriptoren, die dieser Prozess geöffnet haben kann. Einheit: keine

Metrikname	Verfügbar am	Beschreibung
rlimit_num_fds_soft	Linux	Das weiche Limit für die maximale Anzahl von Dateidesk riptoren, die dieser Prozess geöffnet haben kann. Einheit: keine
write_count	Linux, Windows Server	Die Anzahl der Festplatt enschreib vorgänge, die der Prozess ausgeführt hat. Einheit: keine
involuntary_contex t_switches	Linux	Die Anzahl der unfreiwilligen Kontextwe chsel des Prozesses. Einheit: keine
voluntary_context_ switches	Linux	Die Anzahl der freiwilli gen Kontextwe chsel des Prozesses. Einheit: keine

Metrikname	Verfügbar am	Beschreibung
`realtime_priority`	Linux	Die aktuelle Nutzung der Echtzeit-Priorität für den Prozess. Einheit: keine
`nice_priority`	Linux	Die aktuelle Verwendung angenehmer Priorität für den Prozess. Einheit: keine
`signals_pending`	Linux	Die Anzahl der Signale, die noch ausstehen, um vom Prozess verarbeitet zu werden. Einheit: keine
`rlimit_cpu_time_hard`	Linux	Die harte CPU-Zeitressourcen begrenzung für den Prozess. Einheit: keine

Metrikname	Verfügbar am	Beschreibung
`rlimit_cpu_time_soft`	Linux	Die weiche CPU-Zeitr essourcen begrenzung für den Prozess. Einheit: keine
`rlimit_file_locks_hard`	Linux	Die harte Dateisper ren-Resso urcenbegr enzung für den Prozess. Einheit: keine
`rlimit_file_locks_soft`	Linux	Die weiche Dateisper ren-Resso urcenbegr enzung für den Prozess. Einheit: keine
`rlimit_memory_data_hard`	Linux	Die harte Ressource nbegrenzung im Prozess für den Speicher, der für Daten verwendet wird. Einheit: Byte

Metrikname	Verfügbar am	Beschreibung
`rlimit_memory_data_soft`	Linux	Die weiche Ressource nbegrenzung im Prozess für den Speicher, der für Daten verwendet wird. Einheit: Byte
`rlimit_memory_lock ed_hard`	Linux	Die harte Ressource nbegrenzung im Prozess für gesperrten Speicher. Einheit: Byte
`rlimit_memory_lock ed_soft`	Linux	Die weiche Ressource nbegrenzung im Prozess für gesperrten Speicher. Einheit: Byte

Metrikname	Verfügbar am	Beschreibung
`rlimit_memory_rss_hard`	Linux	Die harte Ressource nbegrenzung im Prozess für physischen Speicher. Einheit: Byte
`rlimit_memory_rss_soft`	Linux	Die weiche Ressource nbegrenzung im Prozess für physischen Speicher. Einheit: Byte
`rlimit_memory_stac k_hard`	Linux	Die harte Ressource nbegrenzung im Prozessst apel. Einheit: Byte
`rlimit_memory_stac k_soft`	Linux	Die weiche Ressource nbegrenzung im Prozessst apel. Einheit: Byte

Metrikname	Verfügbar am	Beschreibung
`rlimit_memory_vms_hard`	Linux	Die harte Ressource nbegrenzung im Prozess für virtuellen Speicher. Einheit: Byte
`rlimit_memory_vms_soft`	Linux	Die weiche Ressource nbegrenzung im Prozess für virtuellen Speicher. Einheit: Byte

Vom CloudWatch Agenten importierte Prozessmetriken anzeigen

Nachdem Sie Prozessmetriken in importiert haben CloudWatch, können Sie diese Metriken als Zeitreihendiagramme anzeigen und Alarme erstellen, die diese Metriken überwachen und Sie benachrichtigen, wenn sie einen von Ihnen angegebenen Schwellenwert überschreiten. Das folgende Verfahren zeigt, wie Sie Prozess-Metriken als Zeitreihendiagramm anzeigen. Weitere Informationen zum Einrichten eines -Alarms finden Sie unter CloudWatch Amazon-Alarme verwenden.

Um Prozessmetriken in der CloudWatch Konsole anzuzeigen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie den Namespace für die vom Agent zu erfassenden Metriken. Standardmäßig ist dies CWAgent, aber Sie haben möglicherweise einen anderen Namespace in der CloudWatch Agentenkonfigurationsdatei angegeben.
4. Wählen Sie eine Metrikdimension aus (z. B. Per-Instance Metrics (Metriken pro Instance)).

5. Die Registerkarte All metrics zeigt alle Metriken für diese Dimension im Namespace an. Sie haben die folgenden Möglichkeiten:

   a. Um eine Metrik grafisch darzustellen, müssen Sie das Kontrollkästchen neben der Metrik aktivieren. Um alle Metriken auszuwählen, aktivieren Sie das Kontrollkästchen in der Kopfzeile der Tabelle.

   b. Um die Tabelle sortieren, verwenden Sie die Spaltenüberschrift.

   c. Um nach Ressource zu filtern, müssen Sie zunächst die Ressourcen-ID und dann die Option Zu Suche hinzufügen auswählen.

   d. Um nach Metrik zu filtern, müssen Sie den Metriknamen und anschließend Add to search (Zur Suche hinzufügen) auswählen.

6. (Optional) Um dieses Diagramm zu einem CloudWatch Dashboard hinzuzufügen, wählen Sie Aktionen, Zum Dashboard hinzufügen aus.

Abrufen benutzerdefinierter Metriken mit StatsD

Sie können zusätzliche benutzerdefinierte Metriken aus Ihren Anwendungen oder Diensten abrufen, indem Sie den CloudWatch Agenten mit dem `StatsD` Protokoll verwenden. StatSD ist eine beliebte Open-Source-Lösung, mit der Metriken aus einer Vielzahl von Anwendungen gesammelt werden können. StatSD ist besonders nützlich für das Instrumentieren eigener Metriken. Ein Beispiel für die gemeinsame Verwendung von CloudWatch Agent und StatsD finden Sie unter So überwachen Sie Ihre benutzerdefinierten Anwendungsmetriken mithilfe von Amazon CloudWatch Agent besser.

`StatsD`wird sowohl auf Linux-Servern als auch auf Servern mit Windows Server unterstützt. CloudWatch unterstützt das folgende `StatsD` Format:

```
MetricName:value|type|@sample_rate|#tag1:
  value,tag1...
```

- `MetricName` – Eine Zeichenfolge ohne Doppelpunkte, Balken, #-Zeichen oder @-Zeichen.
- `value` – Dies kann eine Ganzzahl oder Gleitkommazahl sein.
- `type` – Geben Sie `c` für Zähler, `g` für Anzeige, `ms` für Timer, `h` für Histogramm oder `s` für Set an.
- `sample_rate` – (Optional) Eine Gleitkommazahl zwischen 0 und 1. Wird nur für Zähler-, Histogramm- und Timer-Metriken verwendet. Der Standardwert lautet "1" (Erfassung über die gesamte Zeit hinweg).

- `tags`— (Optional) Eine durch Kommas getrennte Liste von Tags. `StatsD`Tags ähneln den Dimensionen in. CloudWatch Verwenden Sie für Schlüssel/Wert-Tags Doppelpunkte, z. B. `env:prod`.

Sie können jeden `StatsD` Client verwenden, der dieses Format verwendet, um die Metriken an den CloudWatch Agenten zu senden. Weitere Informationen zu einigen der verfügbaren `StatsD` Clients finden Sie auf der StatsD-Client-Seite unter GitHub.

Um diese benutzerdefinierten Metriken zu erfassen, fügen Sie die Zeile `"statsd": {}` zum Abschnitt `metrics_collected` der Agentenkonfigurationsdatei hinzu. Sie können diese Zeile manuell hinzufügen. Wenn Sie zum Erstellen der Konfigurationsdatei den Assistenten verwenden, geschieht dies automatisch. Weitere Informationen finden Sie unter Erstellen Sie die CloudWatch Agent-Konfigurationsdatei.

Die `StatsD`-Standardkonfiguration eignet sich für die meisten Benutzer. Es gibt optionale Felder, die Sie ganz nach Bedarf zum Abschnitt statsd der Agentenkonfigurationsdatei hinzufügen können:

- `service_address`— Die Dienstadresse, auf die der CloudWatch Agent hören soll. Das Format ist *`ip`*`:`*`port`*. Wenn Sie die IP-Adresse weglassen, überwacht der Agent alle verfügbaren Schnittstellen. Da nur das UDP-Format unterstützt wird, müssen Sie kein UDP-Präfix anzugeben.

  Der Standardwert ist `:8125`.
- `metrics_collection_interval` – Wie oft (in Sekunden) das `StatsD`-Plug-in ausgeführt wird und Metriken erfasst. Der Standardwert liegt bei 10 Sekunden. Der Bereich liegt zwischen 1 und 172.000.
- `metrics_aggregation_interval`— Wie oft (in Sekunden) werden CloudWatch Metriken zu einzelnen Datenpunkten zusammengefasst. Der Standardwert liegt bei 60 Sekunden.

  Wenn beispielsweise 10 und 60 `metrics_collection_interval` `metrics_aggregation_interval` ist, werden alle 10 Sekunden Daten CloudWatch erfasst. Nach jeder Minute werden die sechs Messwerte dieser Minute zu einem einzigen Datenpunkt zusammengefasst, an CloudWatch den gesendet wird.

  Der Bereich liegt zwischen 0 und 172.000. Wenn für `metrics_aggregation_interval` "0" festgelegt wird, ist die Aggregation von `StatsD`-Metriken deaktiviert.
- `allowed_pending_messages` – Die Anzahl der UDP-Nachrichten, die in die Warteschlange aufgenommen werden dürfen. Wenn die Warteschlange voll ist, beginnt der StatsD-Server, Pakete zu verwerfen. Der Standardwert lautet 10.000.

- `drop_original_metrics` Optional. Wenn Sie das `aggregation_dimensions`-Feld im `metrics`-Abschnitt verwenden, um Metriken zu aggregierten Ergebnissen zusammenzufassen, dann sendet der Agent standardmäßig sowohl die aggregierten Metriken als auch die ursprünglichen Metriken, die für jeden Wert der Dimension getrennt sind. Wenn Sie nicht möchten, dass die ursprünglichen Messwerte an gesendet werden CloudWatch, können Sie diesen Parameter mit einer Liste von Metriken angeben. Für die zusammen mit diesem Parameter angegebenen Metriken werden keine Kennzahlen nach Dimension gemeldet CloudWatch. Stattdessen werden nur die aggregierten Metriken gemeldet. Dadurch verringert sich die Anzahl der Metriken, die der Agent erfasst, was Ihre Kosten senkt.

Im Folgenden finden Sie ein Beispiel für den Abschnitt statsd der Agent-Konfigurationsdatei unter Verwendung des Standard-Ports und von benutzerdefinierten Sammlungs- und Aggregationsintervallen.

```
{
   "metrics":{
      "metrics_collected":{
         "statsd":{
            "service_address":":8125",
            "metrics_collection_interval":60,
            "metrics_aggregation_interval":300
         }
      }
   }
}
```

Vom Agenten importierte StatsD-Metriken anzeigen CloudWatch

Nach dem Import von StatsD-Metriken in CloudWatch können Sie diese Metriken als Zeitreihendiagramme anzeigen und Alarme erstellen, die diese Metriken überwachen und Sie benachrichtigen können, wenn sie einen von Ihnen angegebenen Schwellenwert überschreiten. Das folgende Verfahren zeigt, wie Sie StatsD-Metriken als Zeitreihendiagramm anzeigen. Weitere Informationen zum Einrichten eines -Alarms finden Sie unter CloudWatch Amazon-Alarme verwenden.

So zeigen Sie StatsD-Metriken in der Konsole an CloudWatch

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.

3. Wählen Sie den Namespace für die vom Agent zu erfassenden Metriken. Standardmäßig ist dies CWAgent, aber Sie haben möglicherweise einen anderen Namespace in der CloudWatch Agentenkonfigurationsdatei angegeben.
4. Wählen Sie eine Metrikdimension aus (z. B. Per-Instance Metrics (Metriken pro Instance)).
5. Die Registerkarte All metrics zeigt alle Metriken für diese Dimension im Namespace an. Sie haben die folgenden Möglichkeiten:

   a. Um eine Metrik grafisch darzustellen, müssen Sie das Kontrollkästchen neben der Metrik aktivieren. Um alle Metriken auszuwählen, aktivieren Sie das Kontrollkästchen in der Kopfzeile der Tabelle.

   b. Um die Tabelle sortieren, verwenden Sie die Spaltenüberschrift.

   c. Um nach Ressource zu filtern, müssen Sie zunächst die Ressourcen-ID und dann die Option Zu Suche hinzufügen auswählen.

   d. Um nach Metrik zu filtern, müssen Sie den Metriknamen und anschließend Add to search (Zur Suche hinzufügen) auswählen.
6. (Optional) Um dieses Diagramm zu einem CloudWatch Dashboard hinzuzufügen, wählen Sie Aktionen, Zum Dashboard hinzufügen aus.

Abrufen benutzerdefinierter Metriken mit collectd

Sie können zusätzliche Metriken aus Ihren Anwendungen oder Diensten abrufen, indem Sie den CloudWatch Agenten mit dem Collectd-Protokoll verwenden, das nur auf Linux-Servern unterstützt wird. collectd ist eine beliebte Open-Source-Lösung mit Plugins, die Systemstatistiken für eine Vielzahl von Anwendungen sammeln können. Durch die Kombination der Systemmetriken, die der CloudWatch Agent bereits erfassen kann, mit den zusätzlichen Metriken von collectd können Sie Ihre Systeme und Anwendungen besser überwachen, analysieren und Fehler beheben. Weitere Informationen zu collectd finden Sie unter collectd – Daemon für die Systemstatistikerfassung.

Sie verwenden die gesammelte Software, um die Metriken an den Agenten zu senden. CloudWatch Bei den gesammelten Metriken fungiert der CloudWatch Agent als Server, während das Collectd-Plugin als Client fungiert.

Die collectd-Software wird nicht auf jedem Server automatisch installiert. Führen Sie auf einem Server mit Amazon Linux 2 die folgenden Schritte aus, um collectd zu installieren.

```
sudo amazon-linux-extras install collectd
```

Informationen zum Installieren von collectd auf anderen Systemen finden Sie auf der Download-Seite für collectd.

Um diese benutzerdefinierten Metriken zu erfassen, fügen Sie die Zeile "collectd": {} zum Abschnitt metrics_collected der Agentenkonfigurationsdatei hinzu. Sie können diese Zeile manuell hinzufügen. Wenn Sie zum Erstellen der Konfigurationsdatei den Assistenten verwenden, geschieht dies automatisch. Weitere Informationen finden Sie unter Erstellen Sie die CloudWatch Agent-Konfigurationsdatei.

Optionale Parameter sind ebenfalls verfügbar. Wenn Sie bei Einsatz des collectd-Protokolls nicht `/etc/collectd/auth_file` als Wert für collectd_auth_file verwenden, müssen Sie einige dieser Optionen selbst festlegen.

- service_address: Die Dienstadresse, auf die der CloudWatch Agent hören soll. Das Format ist `"udp://`*`ip`*`:`*`port`*. Der Standardwert ist `udp://127.0.0.1:25826`.
- name_prefix: Ein Präfix zum Anfügen an den Anfang des Namens einer jeden collectd-Metrik. Der Standardwert ist `collectd_`. Die maximale Länge beträgt 255 Zeichen.
- collectd_security_level: Legt die Sicherheitsstufe für die Netzwerkkommunikation fest. Der Standardwert lautet encrypt (Verschlüsseln).

  encrypt (Verschlüsseln) gibt an, dass nur verschlüsselte Daten akzeptiert werden. sign (Signieren) gibt an, dass nur signierte und verschlüsselte Daten akzeptiert werden. none (Keine) gibt an, dass alle Daten akzeptiert werden. Wenn Sie einen Wert für collectd_auth_file angeben, werden verschlüsselte Daten, falls möglich, entschlüsselt.

  Weitere Informationen finden Sie unter Client-Einrichtung und Mögliche Interaktionen in der collectd Wiki.
- collectd_auth_file Legt eine Datei fest, in der Benutzernamen Passwörtern zugeordnet sind. Diese Passwörter werden verwendet, um Signaturen zu verifizieren und verschlüsselte Netzwerkpakete zu entschlüsseln. Sofern angegeben, werden signierte Daten verifiziert und verschlüsselte Pakete entschlüsselt. Andernfalls werden signierte Daten ohne Überprüfung der Signatur akzeptiert und verschlüsselte Daten können nicht entschlüsselt werden.

  Der Standardwert ist `/etc/collectd/auth_file`.

  Wenn collectd_security_level auf none (Keine) gesetzt ist, ist dies optional. Wenn Sie collectd_security_level auf `encrypt` oder sign (Signieren) einstellen, müssen Sie einen Wert für collectd_auth_file angeben.

Bei dem Format der auth-Datei ist jede Zeile ein Benutzernamen, gefolgt von einem Doppelpunkt und einer beliebigen Anzahl von Leerzeichen, gefolgt von dem Passwort. Zum Beispiel:

```
user1: user1_password

user2: user2_password
```

- collectd_typesdb: Eine Liste von einer oder mehreren Dateien, die die Beschreibungen der Datensätze enthalten. Die Liste muss in eckigen Klammern stehen, auch wenn sie nur einen Eintrag enthält. Jeder Eintrag in der Liste muss in Anführungszeichen stehen. Wenn mehrere Einträge vorhanden sind, trennen Sie sie durch Kommas voneinander. Der Standardwert auf Linux-Servern ist `["/usr/share/collectd/types.db"]`. Die Standardeinstellung auf macOs Computern hängt von der Version von collectd ab. z. B. `["/usr/local/Cellar/collectd/5.12.0/share/collectd/types.db"]`.

  Weitere Informationen finden Sie unter https://www.collectd.org/documentation/manpages/types.db.html.
- metrics_aggregation_interval: Wie oft in Sekunden Metriken zu einzelnen Datenpunkten aggregiert werden. CloudWatch Standardmäßig ist ein Zeitraum von 60 Sekunden festgelegt. Der Bereich liegt zwischen 0 und 172,000. Wenn für ihn "0" festgelegt wird, ist die Aggregation von collectd-Metriken deaktiviert.

Es folgt ein Beispiel des collectd-Abschnitts der Agenten-Konfigurationsdatei.

```
{
   "metrics":{
      "metrics_collected":{
         "collectd":{
            "name_prefix":"My_collectd_metrics_",
            "metrics_aggregation_interval":120
         }
      }
   }
}
```

Gesammelte Metriken anzeigen, CloudWatch die vom Agenten importiert wurden

Nachdem Sie die gesammelten Metriken in importiert haben CloudWatch, können Sie diese Metriken als Zeitreihendiagramme anzeigen und Alarme erstellen, die diese Metriken überwachen und Sie

benachrichtigen, wenn sie einen von Ihnen festgelegten Schwellenwert überschreiten. Das folgende Verfahren zeigt, wie Sie Collectd-Metriken als Zeitreihendiagramm anzeigen. Weitere Informationen zum Einrichten eines -Alarms finden Sie unter CloudWatch Amazon-Alarme verwenden.

Um gesammelte Metriken in der Konsole anzuzeigen CloudWatch

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie den Namespace für die vom Agent zu erfassenden Metriken. Standardmäßig ist dies CWAgent, aber Sie haben möglicherweise einen anderen Namespace in der CloudWatch Agentenkonfigurationsdatei angegeben.
4. Wählen Sie eine Metrikdimension aus (z. B. Per-Instance Metrics (Metriken pro Instance)).
5. Die Registerkarte All metrics zeigt alle Metriken für diese Dimension im Namespace an. Sie haben die folgenden Möglichkeiten:

    a. Um eine Metrik grafisch darzustellen, müssen Sie das Kontrollkästchen neben der Metrik aktivieren. Um alle Metriken auszuwählen, aktivieren Sie das Kontrollkästchen in der Kopfzeile der Tabelle.

    b. Um die Tabelle sortieren, verwenden Sie die Spaltenüberschrift.

    c. Um nach Ressource zu filtern, müssen Sie zunächst die Ressourcen-ID und dann die Option Zu Suche hinzufügen auswählen.

    d. Um nach Metrik zu filtern, müssen Sie den Metriknamen und anschließend Add to search (Zur Suche hinzufügen) auswählen.
6. (Optional) Um dieses Diagramm zu einem CloudWatch Dashboard hinzuzufügen, wählen Sie Aktionen, Zum Dashboard hinzufügen aus.

Einrichten und Konfigurieren der Prometheus-Metrikensammlung in Amazon-EC2-Instances

In den folgenden Abschnitten wird erklärt, wie der CloudWatch Agent mit Prometheus-Überwachung auf EC2-Instances installiert wird und wie der Agent so konfiguriert wird, dass er zusätzliche Ziele scannt. Es bietet auch Tutorials zum Einrichten von Beispiel-Workloads für die Verwendung von Tests mit Prometheus-Überwachung.

Informationen zu den vom Agenten unterstützten Betriebssystemen finden Sie unter CloudWatch Erfassen Sie mit dem CloudWatch Agenten Metriken, Logs und Traces

Anforderungen an VPC-Sicherheitsgruppen

Wenn Sie eine VPC verwenden, gelten folgende Anforderungen.

- Die Eingangsregeln der Sicherheitsgruppen für die Prometheus-Workloads müssen die Prometheus-Ports für den CloudWatch Agenten öffnen, damit er die Prometheus-Metriken über die private IP scrapen kann.
- Die Ausgangsregeln der Sicherheitsgruppe für den CloudWatch Agenten müssen es dem CloudWatch Agenten ermöglichen, über eine private IP eine Verbindung zum Port der Prometheus-Workloads herzustellen.

Themen

Schritt 1: Installieren Sie den Agenten CloudWatch

Der erste Schritt besteht darin, den CloudWatch Agenten auf der EC2-Instance zu installieren. Anweisungen finden Sie unter Den CloudWatch Agenten installieren.

Schritt 2: Prometheus-Quellen scrapen und Metriken importieren

Der CloudWatch Agent mit Prometheus-Überwachung benötigt zwei Konfigurationen, um die Prometheus-Metriken zu erfassen. Er folgt der standardmäßigen Prometheus-Konfiguration, wie in <scrape_config> in der Prometheus-Dokumentation erläutert. Die andere ist für die Agentenkonfiguration vorgesehen. CloudWatch

Prometheus-Scrape-Konfiguration

Der CloudWatch Agent unterstützt die standardmäßigen Prometheus-Scrape-Konfigurationen, wiehttps://prometheus.io/docs/prometheus/latest/configuration/configuration/#scrape_config <scrape_config>in der Prometheus-Dokumentation dokumentiert. Sie können diesen Abschnitt bearbeiten, um die Konfigurationen zu aktualisieren, die sich bereits in dieser Datei befinden, und zusätzliche Prometheus-Scraping-Ziele hinzufügen. Eine Beispielkonfigurationsdatei enthält die folgenden globalen Konfigurationszeilen:

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus.yaml
global:
```

```
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
- job_name: MY_JOB
  sample_limit: 10000
  file_sd_configs:
    - files: ["C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\prometheus_sd_1.yaml",
 "C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\prometheus_sd_2.yaml"]
```

Der Abschnitt `global` gibt Parameter an, die in allen Konfigurationskontexten gültig sind. Sie fungieren auch als Standardwerte für andere Konfigurationsabschnitte. Es enthält die folgenden Parameter:

- `scrape_interval` – Definiert, wie oft das Scraping von zielen durchgeführt werden soll.
- `scrape_timeout` – Definiert, wie lange gewartet werden soll, bis für eine Scrape-Anforderung eine Zeitüberschreitung eintritt.

`scrape_configs` gibt einen Satz von Zielen und Parametern an, mit denen festgelegt wird, wie sie das Scraping durchführen sollen. Es enthält die folgenden Parameter:

- `job_name`— Der Auftragsname, der standardmäßig gescrapten Metriken zugewiesen ist.
- `sample_limit` – Pro-Scrape-Limit für die Anzahl der Scraping-Proben, die akzeptiert werden.
- `file_sd_configs` – Liste der Konfigurationen für die Dateiserviceerkennung. Es liest eine Reihe von Dateien, die eine Liste von null oder mehr statischen Konfigurationen enthalten. Der Abschnitt `file_sd_configs` enthält einen Parameter `files`, der Muster für Dateien definiert, aus denen Zielgruppen extrahiert werden.

Der CloudWatch Agent unterstützt die folgenden Service Discovery-Konfigurationstypen.

`static_config` Ermöglicht die Angabe einer Liste von Zielen und eines gemeinsamen Beschriftungssatzes für diese. Es ist der kanonische Weg, statische Ziele in einer Scrape-Konfiguration anzugeben.

Im Folgenden finden Sie eine statische Beispielkonfiguration, um Prometheus Metriken von einem lokalen Host zu scrapen. Metriken können auch von anderen Servern gescrapet werden, wenn der Prometheus-Port für den Server geöffnet ist, auf dem der Agent ausgeführt wird.

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus_sd_1.yaml
- targets:
```

```
    - 127.0.0.1:9404
  labels:
    key1: value1
    key2: value2
```

Dieses Beispiel enthält die folgenden Parameter:

- `targets` – Die Ziele, die von der statischen Konfiguration gescrapet werden.
- `labels` – Beschriftungen, die allen Metriken zugewiesen sind, die von den Zielen gescrapet werden.

`ec2_sd_config` Ermöglicht das Abrufen von Scrape-Zielen von Amazon-EC2-Instances. Im Folgenden finden Sie ein `ec2_sd_config`-Beispiel zum Scrapen von Prometheus-Metriken aus einer Liste von EC2-Instances. Die Prometheus-Ports dieser Instanzen müssen für den Server geöffnet werden, auf dem der CloudWatch Agent ausgeführt wird. Die IAM-Rolle für die EC2-Instance, auf der der CloudWatch Agent ausgeführt wird, muss die Berechtigung enthalten. `ec2:DescribeInstance` Sie könnten beispielsweise die verwaltete Richtlinie AmazonEC2 an die Instance anhängen, ReadOnlyAccess auf der der Agent ausgeführt wird. CloudWatch

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus.yaml
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: MY_JOB
    sample_limit: 10000
    ec2_sd_configs:
      - region: us-east-1
        port: 9404
        filters:
          - name: instance-id
            values:
              - i-98765432109876543
              - i-12345678901234567
```

Dieses Beispiel enthält die folgenden Parameter:

- `region`— Die AWS Region, in der sich die EC2-Zielinstanz befindet. Wenn Sie dieses Feld leer lassen, wird die Region aus den Instance-Metadaten verwendet.
- `port` – Der Port, von dem Metriken gescrapet werden.

- `filters` – Optionale Filter zum Filtern der Instanceliste. Dieses Beispiel filtert basierend auf EC2-Instance-IDs. Weitere Kriterien, nach denen Sie filtern können, finden Sie unter DescribeInstances.

CloudWatch Agentenkonfiguration für Prometheus

Die CloudWatch Agenten-Konfigurationsdatei enthält `prometheus` Abschnitte sowohl unter als auch`logs`. `metrics_collected` Es enthält die folgenden Parameter.

- Clustername – Gibt den Clusternamen an, der als Bezeichnung im Protokollereignis hinzugefügt werden soll. Dies ist ein optionales Feld.
- log_group_name – Gibt den Namen der Protokollgruppe für die Prometheus-Scrape-Metriken an.
- prometheus_config_path – gibt den Pfad der Prometheus-Scrape-Konfigurationsdatei an.
- emf_processor – Gibt die Prozessorkonfiguration im eingebetteten Metrikformat an. Weitere Hinweise zum eingebetteten Metrik-Format finden Sie unter Einbetten von Metriken in Protokollen.

  Der Abschnitt `emf_processor` kann die folgenden Parameter enthalten:

  - metric_declaration_dedup – Wenn es auf „true“ gesetzt ist, wird die Deduplizierungsfunktion für die eingebetteten Metrikformatmetriken aktiviert.
  - metric_namespace — Gibt den Metrik-Namespace für die ausgegebenen Metriken an. CloudWatch
  - metric_unit – Gibt die Metrikname:Metrikeinheitszuordnung an. Hinweise zu unterstützten Metrik-Einheiten finden Sie unter. MetricDatum
  - metric_declaration – sind Abschnitte, die das Array von Protokollen mit eingebettetem Metrikformat angeben, das generiert werden soll. Für jede Prometheus-Quelle, aus der der CloudWatch Agent standardmäßig importiert, gibt es `metric_declaration` Abschnitte. Diese Abschnitte enthalten jeweils die folgenden Felder:
    - `source_labels` gibt den Wert der Beschriftungen an, die von der `label_matcher`-Zeile überprüft werden.
    - `label_matcher` ist ein regulärer Ausdruck, der den Wert der in `source_labels` aufgelisteten Beschriftungen überprüft. Die übereinstimmenden Metriken werden für die Aufnahme in das eingebettete Metrikformat aktiviert, an das gesendet wird. CloudWatch
    - `metric_selectors`ist ein regulärer Ausdruck, der die Metriken angibt, die gesammelt und an sie gesendet werden sollen CloudWatch.
    - `dimensions`ist die Liste der Labels, die als CloudWatch Dimensionen für jede ausgewählte Metrik verwendet werden sollen.

Im Folgenden finden Sie ein Beispiel für eine CloudWatch Agentenkonfiguration für Prometheus.

```
{
   "logs":{
      "metrics_collected":{
         "prometheus":{
            "cluster_name":"prometheus-cluster",
            "log_group_name":"Prometheus",
            "prometheus_config_path":"C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
\prometheus.yaml",
            "emf_processor":{
               "metric_declaration_dedup":true,
               "metric_namespace":"CWAgent-Prometheus",
               "metric_unit":{
                  "jvm_threads_current": "Count",
                  "jvm_gc_collection_seconds_sum": "Milliseconds"
               },
               "metric_declaration":[
                  {
                     "source_labels":[
                        "job", "key2"
                     ],
                     "label_matcher":"MY_JOB;^value2",
                     "dimensions":[
                        [
                           "key1", "key2"
                        ],
                        [
                           "key2"
                        ]
                     ],
                     "metric_selectors":[
                        "^jvm_threads_current$",
                        "^jvm_gc_collection_seconds_sum$"
                     ]
                  }
               ]
            }
         }
      }
   }
}
```

Im vorherigen Beispiel wird ein eingebetteter Metrikformatabschnitt konfiguriert, der als Protokollereignis gesendet wird, wenn die folgenden Bedingungen erfüllt sind:

- Der Wert der Beschriftung `job` ist `MY_JOB`
- Der Wert der Beschriftung `key2` ist `value2`
- Die Prometheus-Metriken `jvm_threads_current` und `jvm_gc_collection_seconds_sum` enthalten sowohl `job`-als auch `key2`-Beschriftungen.

Das Protokollereignis, das gesendet wird, enthält den folgenden hervorgehobenen Abschnitt.

```
{
    "CloudWatchMetrics": [
        {
            "Metrics": [
                {
                    "Unit": "Count",
                    "Name": "jvm_threads_current"
                },
                {
                    "Unit": "Milliseconds",
                    "Name": "jvm_gc_collection_seconds_sum"
                }
            ],
            "Dimensions": [
                [
                    "key1",
                    "key2"
                ],
                [
                    "key2"
                ]
            ],
            "Namespace": "CWAgent-Prometheus"
        }
    ],
    "ClusterName": "prometheus-cluster",
    "InstanceId": "i-0e45bd06f196096c8",
    "Timestamp": "1607966368109",
    "Version": "0",
    "host": "EC2AMAZ-PDDOIUM",
    "instance": "127.0.0.1:9404",
```

```
    "jvm_threads_current": 2,
    "jvm_gc_collection_seconds_sum": 0.006000000000000002,
    "prom_metric_type": "gauge",
    ...
}
```

Beispiel: Einrichten von Java/JMX-Beispiel-Workloads für Prometheus-Metriktests

JMX Exporter ist ein offizieller Prometheus-Exporter, der JMX mBeans als Prometheus-Metriken erfassen und verfügbar machen kann. Weitere Informationen finden Sie unter prometheus/jmx_exporter.

Der CloudWatch Agent kann vordefinierte Prometheus-Metriken von Java Virtual Machine (JVM), Hjava und Tomcat (Catalina) von einem JMX-Exporter auf EC2-Instances sammeln.

CloudWatch Schritt 1: Installieren Sie den Agenten

Der erste Schritt besteht darin, den CloudWatch Agenten auf der EC2-Instance zu installieren. Anweisungen finden Sie unter Den CloudWatch Agenten installieren.

Schritt 2: Starten der Java/JMX-Workload

Der nächste Schritt besteht darin, die Java/JMX-Workload zu starten.

Laden Sie zunächst die neueste JMX-Exporter-JAR-Datei vom folgenden Speicherort herunter: prometheus/jmx_exporter.

Verwenden des .jar für Ihre Beispielanwendung

Die Beispielbefehle in den folgenden Abschnitten verwenden `SampleJavaApplication-1.0-SNAPSHOT.jar` als JAR-Datei. Ersetzen Sie diese Teile der Befehle durch die JAR-Datei für Ihre Anwendung.

Vorbereiten der JMX-Exporter-Konfiguration

Die `config.yaml`-Datei ist die JMX-Exporter-Konfigurationsdatei. Weitere Informationen finden Sie unter Konfiguration in der JMX-Exporter-Dokumentation.

Hier ist eine Beispielkonfiguration für Java und Tomcat.

```
---
lowercaseOutputName: true
lowercaseOutputLabelNames: true
```

```
rules:
- pattern: 'java.lang<type=OperatingSystem><>(FreePhysicalMemorySize|
TotalPhysicalMemorySize|FreeSwapSpaceSize|TotalSwapSpaceSize|SystemCpuLoad|
ProcessCpuLoad|OpenFileDescriptorCount|AvailableProcessors)'
  name: java_lang_OperatingSystem_$1
  type: GAUGE

- pattern: 'java.lang<type=Threading><>(TotalStartedThreadCount|ThreadCount)'
  name: java_lang_threading_$1
  type: GAUGE

- pattern: 'Catalina<type=GlobalRequestProcessor, name=\"(\w+-\w+)-(\d+)\"><>(\w+)'
  name: catalina_globalrequestprocessor_$3_total
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina global $3
  type: COUNTER

- pattern: 'Catalina<j2eeType=Servlet, WebModule=//([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-
a-zA-Z0-9+&@#/%=~_|]), name=([-a-zA-Z0-9+/$%~_-|!.]*), J2EEApplication=none,
 J2EEServer=none><>(requestCount|maxTime|processingTime|errorCount)'
  name: catalina_servlet_$3_total
  labels:
    module: "$1"
    servlet: "$2"
  help: Catalina servlet $3 total
  type: COUNTER

- pattern: 'Catalina<type=ThreadPool, name="(\w+-\w+)-(\d+)"><>(currentThreadCount|
currentThreadsBusy|keepAliveCount|pollerThreadCount|connectionCount)'
  name: catalina_threadpool_$3
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina threadpool $3
  type: GAUGE

- pattern: 'Catalina<type=Manager, host=([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-a-zA-
Z0-9+&@#/%=~_|]), context=([-a-zA-Z0-9+/$%~_-|!.]*)><>(processingTime|sessionCounter|
rejectedSessions|expiredSessions)'
  name: catalina_session_$3_total
  labels:
    context: "$2"
```

```
    host: "$1"
  help: Catalina session $3 total
  type: COUNTER

- pattern: ".*"
```

Starten Sie die Java-Anwendung mit dem Prometheus-Exporter

Starten der Beispielanwendung Dadurch werden Prometheus-Metriken an Port 9404 ausgegeben. Stellen Sie sicher, dass Sie den Einstiegspunkt `com.gubupt.sample.app.App` durch die richtigen Informationen für Ihre Java-Beispielanwendung ersetzen.

Geben Sie unter Linux den folgenden Befehl ein.

```
$ nohup java -javaagent:./jmx_prometheus_javaagent-0.14.0.jar=9404:./config.yaml -cp
  ./SampleJavaApplication-1.0-SNAPSHOT.jar com.gubupt.sample.app.App &
```

Geben Sie unter Windows den folgenden Befehl ein.

```
PS C:\> java -javaagent:.\jmx_prometheus_javaagent-0.14.0.jar=9404:.\config.yaml -cp  .
\SampleJavaApplication-1.0-SNAPSHOT.jar com.gubupt.sample.app.App
```

Überprüfen der Prometheus-Metriken

Stellen Sie sicher, dass Prometheus-Metriken ausgegeben werden.

Geben Sie unter Linux den folgenden Befehl ein.

```
$ curl localhost:9404
```

Geben Sie unter Windows den folgenden Befehl ein.

```
PS C:\> curl  http://localhost:9404
```

Beispielausgabe unter Linux:

```
StatusCode        : 200
StatusDescription : OK
Content           : # HELP jvm_classes_loaded The number of classes that are currently
 loaded in the JVM
                    # TYPE jvm_classes_loaded gauge
                    jvm_classes_loaded 2526.0
```

```
                    # HELP jvm_classes_loaded_total The total number of class...
RawContent        : HTTP/1.1 200 OK
                    Content-Length: 71908
                    Content-Type: text/plain; version=0.0.4; charset=utf-8
                    Date: Fri, 18 Dec 2020 16:38:10 GMT

                    # HELP jvm_classes_loaded The number of classes that are
 currentl...
Forms             : {}
Headers           : {[Content-Length, 71908], [Content-Type, text/plain; version=0.0.4;
 charset=utf-8], [Date, Fri, 18
                    Dec 2020 16:38:10 GMT]}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : System.__ComObject
RawContentLength  : 71908
```

Schritt 3: Den CloudWatch Agenten so konfigurieren, dass er Prometheus-Metriken scannt

Richten Sie als Nächstes die Prometheus-Scrape-Konfiguration in der CloudWatch Agentenkonfigurationsdatei ein.

So richten Sie die Prometheus-Scrape-Konfiguration für das Java/JMX-Beispiel ein

1. Richten Sie die Konfiguration für `file_sd_config`und `static_config` ein.

   Geben Sie unter Linux den folgenden Befehl ein.

   ```
   $ cat /opt/aws/amazon-cloudwatch-agent/var/prometheus.yaml
   global:
     scrape_interval: 1m
     scrape_timeout: 10s
   scrape_configs:
     - job_name: jmx
       sample_limit: 10000
       file_sd_configs:
         - files: [ "/opt/aws/amazon-cloudwatch-agent/var/prometheus_file_sd.yaml" ]
   ```

   Geben Sie unter Windows den folgenden Befehl ein.

   ```
   PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus.yaml
   ```

```
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: jmx
    sample_limit: 10000
    file_sd_configs:
      - files: [ "C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
\prometheus_file_sd.yaml" ]
```

2. Richten Sie die Konfiguration der Scrape-Ziele ein.

   Geben Sie unter Linux den folgenden Befehl ein.

   ```
   $ cat /opt/aws/amazon-cloudwatch-agent/var/prometheus_file_sd.yaml
   - targets:
     - 127.0.0.1:9404
     labels:
       application: sample_java_app
       os: linux
   ```

   Geben Sie unter Windows den folgenden Befehl ein.

   ```
   PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus_file_sd.yaml
   - targets:
     - 127.0.0.1:9404
     labels:
       application: sample_java_app
       os: windows
   ```

3. Richten Sie die Scrape-Konfiguration von Prometheus mit `ec2_sc_config` ein. Ersetzen Sie *`your-ec2-instance-id`* durch die richtige EC2-Instance-ID.

   Geben Sie unter Linux den folgenden Befehl ein.

   ```
   $ cat .\prometheus.yaml
   global:
     scrape_interval: 1m
     scrape_timeout: 10s
   scrape_configs:
     - job_name: jmx
       sample_limit: 10000
   ```

```
    ec2_sd_configs:
      - region: us-east-1
        port: 9404
        filters:
          - name: instance-id
            values:
              - your-ec2-instance-id
```

Geben Sie unter Windows den folgenden Befehl ein.

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus_file_sd.yaml
- targets:
  - 127.0.0.1:9404
  labels:
    application: sample_java_app
    os: windows
```

4. Richten Sie die Agentenkonfiguration ein. CloudWatch Wechseln Sie zunächst in das richtige Verzeichnis. Unter Linux ist es `/opt/aws/amazon-cloudwatch-agent/var/cwagent-config.json`. Unter Windows ist es `C:\ProgramData\Amazon\AmazonCloudWatchAgent\cwagent-config.json`.

   Im Folgenden sehen Sie eine Beispielkonfiguration mit definierten Java/JHX-Prometheus-Metriken. Achten Sie darauf, *path-to-Prometheus-Scrape-Configuration-file* durch den richtigen Pfad zu ersetzen.

```
{
  "agent": {
    "region": "us-east-1"
  },
  "logs": {
    "metrics_collected": {
      "prometheus": {
        "cluster_name": "my-cluster",
        "log_group_name": "prometheus-test",
        "prometheus_config_path": "path-to-Prometheus-Scrape-Configuration-file",
        "emf_processor": {
          "metric_declaration_dedup": true,
          "metric_namespace": "PrometheusTest",
          "metric_unit":{
            "jvm_threads_current": "Count",
            "jvm_classes_loaded": "Count",
```

```
            "java_lang_operatingsystem_freephysicalmemorysize": "Bytes",
            "catalina_manager_activesessions": "Count",
            "jvm_gc_collection_seconds_sum": "Seconds",
            "catalina_globalrequestprocessor_bytesreceived": "Bytes",
            "jvm_memory_bytes_used": "Bytes",
            "jvm_memory_pool_bytes_used": "Bytes"
          },
          "metric_declaration": [
            {
              "source_labels": ["job"],
              "label_matcher": "^jmx$",
              "dimensions": [["instance"]],
              "metric_selectors": [
                "^jvm_threads_current$",
                "^jvm_classes_loaded$",
                "^java_lang_operatingsystem_freephysicalmemorysize$",
                "^catalina_manager_activesessions$",
                "^jvm_gc_collection_seconds_sum$",
                "^catalina_globalrequestprocessor_bytesreceived$"
              ]
            },
            {
              "source_labels": ["job"],
              "label_matcher": "^jmx$",
              "dimensions": [["area"]],
              "metric_selectors": [
                "^jvm_memory_bytes_used$"
              ]
            },
            {
              "source_labels": ["job"],
              "label_matcher": "^jmx$",
              "dimensions": [["pool"]],
              "metric_selectors": [
                "^jvm_memory_pool_bytes_used$"
              ]
            }
          ]
        }
      }
    },
    "force_flush_interval": 5
  }
```

```
}
```

5. Starten Sie den CloudWatch Agenten neu, indem Sie einen der folgenden Befehle eingeben.

   Geben Sie unter Linux den folgenden Befehl ein.

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
   config -m ec2 -s -c file:/opt/aws/amazon-cloudwatch-agent/var/cwagent-config.json
   ```

   Geben Sie unter Windows den folgenden Befehl ein.

   ```
   & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
    -a fetch-config -m ec2 -s -c file:C:\ProgramData\Amazon\AmazonCloudWatchAgent
   \cwagent-config.json
   ```

Anzeigen der Prometheus-Metriken und Protokolle

Sie können nun die erfassten Java/JMX-Metriken anzeigen.

So zeigen Sie die Metriken für Ihren Java/JMX-Beispiel-Workload an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie in der Region, in der Ihr Cluster ausgeführt wird, im linken Navigationsbereich Metriken aus. Suchen Sie den PrometheusTestNamespace, um die Metriken zu sehen.
3. Um die CloudWatch Protokollereignisse anzuzeigen, wählen Sie im Navigationsbereich Protokollgruppen aus. Die Ereignisse befinden sich in der Protokollgruppe prometheus-test.

Installieren Sie den CloudWatch Agenten mithilfe des Amazon CloudWatch Observability EKS-Add-ons

Das Amazon CloudWatch Observability EKS-Add-on installiert den CloudWatch Agenten und den Fluent-Bit-Agenten auf einem Amazon EKS-Cluster, wobei die erweiterte Observability von Container Insights für Amazon EKS und CloudWatch Application Signals standardmäßig aktiviert ist. Mit dem Add-On können Sie Infrastrukturmetriken, Anwendungs-Leistungstelemetrie und Container-Protokolle aus dem Amazon-EKS-Cluster sammeln.

Bei Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS werden die Container-Insights-Metriken pro Beobachtung abgerechnet, anstatt pro gespeicherter Metrik oder

aufgenommenem Protokoll. Bei Application Signals basiert die Abrechnung auf eingehenden Anfragen an Ihre Anwendungen, ausgehenden Anfragen von Ihren Anwendungen und jedem konfigurierten Servicelevel-Ziel (SLO). Jede eingehende Anfrage generiert ein Anwendungssignal, und jede ausgehende Anfrage generiert ein Anwendungssignal. Jeder SLO erzeugt zwei Anwendungssignale pro Messzeitraum. Weitere Informationen zur CloudWatch Preisgestaltung finden Sie unter CloudWatch Amazon-Preise.

Das Amazon EKS-Add-on ermöglicht Container Insights sowohl auf Linux- als auch auf Windows-Worker-Knoten im Amazon EKS-Cluster. Um Container Insights unter Windows zu aktivieren, müssen Sie Version 1.5.0 oder höher des Amazon EKS-Add-ons verwenden. Derzeit wird Application Signals unter Windows in Amazon EKS-Clustern nicht unterstützt.

Das Amazon CloudWatch Observability EKS-Add-on wird auf Amazon EKS-Clustern unterstützt, die mit Kubernetes Version 1.23 oder höher ausgeführt werden.

Wenn Sie das Add-on installieren, müssen Sie auch IAM-Berechtigungen erteilen, damit der CloudWatch Agent Metriken, Protokolle und Traces an senden kann. CloudWatch Es gibt zwei Möglichkeiten dafür:

- Fügen Sie eine Richtlinie an die IAM-Rolle Ihrer Worker-Knoten an. Diese Option gewährt Worker-Knoten die Erlaubnis, Telemetrie an sie zu senden. CloudWatch
- Verwenden Sie eine IAM-Rolle für Servicekonten für die Agenten-Pods und fügen Sie die Richtlinie an diese Rolle an. Dies funktioniert nur für Amazon-EKS-Cluster. Diese Option gewährt nur CloudWatch Zugriff auf die entsprechenden Agenten-Pods.

Option 1: Installation mit IAM-Berechtigungen auf Worker-Knoten

Um diese Methode zu verwenden, fügen Sie zunächst die CloudWatchAgentServerPolicyIAM-Richtlinie Ihren Worker-Knoten hinzu, indem Sie den folgenden Befehl eingeben. Ersetzen Sie *my-worker-node-role*diesen Befehl durch die IAM-Rolle, die von Ihren Kubernetes-Worker-Knoten verwendet wird.

```
aws iam attach-role-policy \
--role-name my-worker-node-role \
--policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
```

Installieren Sie anschließend das Amazon CloudWatch Observability EKS-Add-on. Um das Add-on zu installieren, können Sie die AWS CLI, die Konsole oder AWS CloudFormation Terraform verwenden.

AWS CLI

AWS CLI Um das Amazon CloudWatch Observability EKS-Add-on zu installieren

Geben Sie den folgenden Befehl ein. Ersetzen Sie *my-cluster-name* mit dem Namen Ihres Clusters.

```
aws eks create-addon --addon-name amazon-cloudwatch-observability --cluster-name my-cluster-name
```

Amazon EKS console

So verwenden Sie die Amazon EKS-Konsole, um das Amazon CloudWatch Observability EKS-Add-on hinzuzufügen

1. Öffnen Sie die Amazon-EKS-Konsole unter https://console.aws.amazon.com/eks/home#/clusters.
2. Wählen Sie im linken Navigationsbereich die Option Cluster aus.
3. Wählen Sie den Namen des Clusters, für den Sie das Amazon CloudWatch Observability EKS-Add-on konfigurieren möchten.
4. Wählen Sie die Registerkarte Add-ons.
5. Wählen Sie Weitere Add-Ons erhalten.
6. Führen Sie auf der Seite Add-Ons auswählen die folgenden Schritte aus:
   a. Aktivieren Sie im Bereich Amazon EKS-Addons das Kontrollkästchen Amazon CloudWatch Observability.
   b. Wählen Sie Weiter aus.
7. Gehen Sie auf der Seite Konfigurieren ausgewählter Add-Ons-Einstellungen wie folgt vor:
   a. Wählen Sie die Version aus, die Sie verwenden möchten.
   b. Wählen Sie für IAM-Rolle auswählen die Option Vom Knoten erben aus
   c. (Optional) Sie können Optionale Konfigurationseinstellungen erweitern. Wenn Sie Überschreiben für Methode zur Konfliktlösung auswählen, können eine oder mehrere

der Einstellungen für das vorhandene Add-on mit den Einstellungen des Amazon-EKS-Add-ons überschrieben werden. Wenn Sie diese Option nicht aktivieren und ein Konflikt mit Ihren vorhandenen Einstellungen vorliegt, schlägt der Vorgang fehl. Sie können die sich daraus ergebende Fehlermeldung heranziehen, um den Konflikt zu beheben. Stellen Sie vor der Auswahl dieser Option sicher, dass das Amazon-EKS-Add-on keine Einstellungen verwaltet, die Sie selbst verwalten müssen.

d. Wählen Sie Weiter aus.

8. Wählen Sie auf der Seite Überprüfen und hinzufügen die Option Erstellen aus. Nachdem die Installation der Add-Ons abgeschlossen ist, wird Ihr installiertes Add-On angezeigt.

AWS CloudFormation

AWS CloudFormation Zur Installation des Amazon CloudWatch Observability EKS-Add-ons

Ersetzen Sie *my-cluster-name* mit dem Namen Ihres Clusters. Weitere Informationen finden Sie unter AWS::EKS::Addon.

```
{
    "Resources": {
        "EKSAddOn": {
            "Type": "AWS::EKS::Addon",
            "Properties": {
                "AddonName": "amazon-cloudwatch-observability",
                "ClusterName": "my-cluster-name"
            }
        }
    }
}
```

Terraform

Um Terraform zur Installation des Amazon CloudWatch Observability EKS-Add-ons zu verwenden

Ersetzen Sie *my-cluster-name* mit dem Namen Ihres Clusters. Weitere Informationen finden Sie unter Resource: aws_eks_addon.

```
resource "aws_eks_addon" "example" {
  addon_name   = "amazon-cloudwatch-observability"
  cluster_name = "my-cluster-name"
```

```
}
```

Option 2: Installation mithilfe der IAM-Servicekontorolle

Vor dem Verwenden dieser Methode, überprüfen Sie die folgenden Voraussetzungen:

- Sie besitzen einen funktionsfähigen Amazon-EKS-Cluster mit angefügten Knoten in einer der AWS-Regionen , von denen Container Insights unterstützt wird. Eine Liste der unterstützten Regionen finden Sie unter Container Insights.
- Sie haben `kubectl` für den Cluster installiert und konfiguriert. Weitere Informationen finden Sie unter Installieren von `kubectl` im Amazon-EKS-Benutzerhandbuch.
- Sie haben `eksctl` installiert. Weitere Informationen finden Sie unter Installieren oder Aktualisieren von `eksctl` im Amazon-EKS-Benutzerhandbuch.

So installieren Sie das Amazon CloudWatch Observability EKS-Add-on mithilfe der IAM-Servicekontorolle

1. Geben Sie den folgenden Befehl ein, um einen OpenID-Connect-Anbieter (OIDC) zu erstellen, falls der Cluster noch keinen hat. Weitere Informationen finden Sie unter Konfigurieren eines Kubernetes-Servicekontos zur Übernahme einer IAM-Rolle im Amazon-EKS-Benutzerhandbuch.

   ```
   eksctl utils associate-iam-oidc-provider --cluster my-cluster-name --approve
   ```

2. Geben Sie den folgenden Befehl ein, um die IAM-Rolle mit der angehängten CloudWatchAgentServerPolicyRichtlinie zu erstellen, und konfigurieren Sie das Agent-Servicekonto so, dass es diese Rolle mithilfe von OIDC übernimmt. *my-cluster-name*Ersetzen Sie es durch den Namen Ihres Clusters und *my-service-account-role*ersetzen Sie es durch den Namen der Rolle, der Sie das Dienstkonto zuordnen möchten. Wenn diese Rolle noch nicht vorhanden ist, erstellt `eksctl` sie für Sie.

   ```
   eksctl create iamserviceaccount \
     --name cloudwatch-agent \
     --namespace amazon-cloudwatch --cluster my-cluster-name \
     --role-name my-service-account-role \
     --attach-policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy \
     --role-only \
     --approve
   ```

3. Installieren Sie das Add-On indem Sie den folgenden Befehl eingeben. *my-cluster-name*Ersetzen Sie es durch den Namen Ihres Clusters, ersetzen Sie *111122223333* durch Ihre Konto-ID und *my-service-account-role*ersetzen Sie es durch die IAM-Rolle, die im vorherigen Schritt erstellt wurde.

```
aws eks create-addon --addon-name amazon-cloudwatch-observability --cluster-
name my-cluster-name --service-account-role-arn arn:aws:iam::111122223333:role/my-
service-account-role
```

(Optional) Zusätzliche Konfiguration

Deaktivieren Sie die Erfassung von Container-Logs

Standardmäßig verwendet das Add-on Fluent Bit, um Container-Logs von allen Pods zu sammeln und sendet die Logs dann an CloudWatch Logs. Informationen darüber, welche Protokolle gesammelt werden, finden Sie unter Einrichten von Fluent Bit.

Um die Erfassung von Container-Logs zu deaktivieren, übergeben Sie beim Erstellen oder Aktualisieren des Add-ons die folgende Option:

```
--configuration-values '{ "containerLogs": { "enabled": false } }'
```

Deaktivieren Sie die Erfassung von NVIDIA-GPU-Metriken

Ab Version 1.300034.0 des CloudWatch Agenten erfasst Container Insights standardmäßig NVIDIA-GPU-Metriken von EKS-Workloads. Diese Metriken sind in der Tabelle unter aufgeführt. NVIDIA GPU-Metriken

Sie können die Erfassung von NVIDIA-GPU-Metriken deaktivieren, indem Sie die `accelerated_compute_metrics` Option in der CloudWatch Agentenkonfigurationsdatei auf setzen`false`. Diese Option befindet sich im `kubernetes` Abschnitt des `metrics_collected` Abschnitts in der CloudWatch Konfigurationsdatei. Im Folgenden finden Sie ein Beispiel für eine Opt-Out-Konfiguration.

```
{
  "agent": {
    "region": "us-east-1"
  },
  "logs": {
```

```
    "metrics_collected": {
      "emf": {
      },
      "kubernetes": {
        "enhanced_container_insights": true,
        "accelerated_compute_metrics": false
      }
    },
    "force_flush_interval": 5,
  }
}
```

Verwenden Sie eine benutzerdefinierte CloudWatch Agentenkonfiguration

Um andere Metriken, Logs oder Traces mithilfe des CloudWatch Agenten zu erfassen, können Sie eine benutzerdefinierte Konfiguration angeben und gleichzeitig Container Insights und CloudWatch Application Signals aktiviert lassen. Betten Sie dazu die CloudWatch Agentenkonfigurationsdatei in den Konfigurationsschlüssel unter dem Agentenschlüssel der erweiterten Konfiguration ein, den Sie bei der Erstellung oder Aktualisierung des EKS-Add-ons verwenden können. Im Folgenden wird die standardmäßige Agentenkonfiguration dargestellt, wenn Sie keine zusätzliche Konfiguration angeben.

> **Important**
>
> Jede benutzerdefinierte Konfiguration, die Sie mithilfe zusätzlicher Konfigurationseinstellungen angeben, hat Vorrang vor der vom Agenten verwendeten Standardkonfiguration. Achten Sie darauf, standardmäßig aktivierte Funktionen wie Container Insights mit verbesserter Beobachtbarkeit und CloudWatch Application Signals nicht ungewollt zu deaktivieren. In dem Szenario, bei dem Sie eine benutzerdefinierte Agentenkonfiguration bereitstellen müssen, empfehlen wir, die folgende Standardkonfiguration als Basiskonfiguration zu verwenden und sie dann entsprechend zu ändern.

```
--configuration-values '{
  "agent": {
    "config": {
      "logs": {
        "metrics_collected": {
          "app_signals": {},
          "kubernetes": {
```

```
            "enhanced_container_insights": true
          }
        }
      },
      "traces": {
        "traces_collected": {
          "app_signals": {}
        }
      }
    }

}'
```

Das folgende Beispiel zeigt die Standard-Agentenkonfiguration für den CloudWatch Agenten unter Windows. Der CloudWatch Agent unter Windows unterstützt keine benutzerdefinierte Konfiguration.

```
{
  "logs": {
    "metrics_collected": {
      "kubernetes": {
        "enhanced_container_insights": true
      },
    }
  }
}
```

Webhook-TLS-Zulassungszertifikate verwalten

Das Amazon CloudWatch Observability EKS-Add-on nutzt Kubernetes-Zulassungswebhooks zur Validierung und Mutation sowie `Instrumentation` benutzerdefinierte Ressourcenanfragen (CR) `AmazonCloudWatchAgent` und optional Kubernetes-Pod-Anfragen auf dem Cluster, wenn Application Signals aktiviert ist. CloudWatch In Kubernetes benötigen Webhooks ein TLS-Zertifikat, dem der API-Server vertraut, um die sichere Kommunikation zu gewährleisten.

Standardmäßig generiert das Amazon CloudWatch Observability EKS-Add-on automatisch eine selbstsignierte CA und ein von dieser CA signiertes TLS-Zertifikat, um die Kommunikation zwischen dem API-Server und dem Webhook-Server zu sichern. Dieses automatisch generierte Zertifikat hat eine Standardablaufzeit von 10 Jahren und wird nach Ablauf nicht automatisch verlängert. Darüber hinaus werden das CA-Paket und das Zertifikat jedes Mal neu generiert, wenn das Add-On aktualisiert oder neu installiert wird, wodurch der Ablauf zurückgesetzt wird. Wenn Sie den Standardablauf des automatisch generierten Zertifikats ändern möchten, können Sie beim Erstellen

oder Aktualisieren des Add-Ons die folgenden zusätzlichen Konfigurationen verwenden. Ersetzen Sie es *expiry-in-days*durch die gewünschte Ablaufdauer in Tagen.

```
--configuration-values '{ "admissionWebhooks": { "autoGenerateCert":
 { "expiryDays": expiry-in-days } } }'
```

Für eine sicherere und featurereichere Zertifizierungsstellen-Lösung bietet das Add-On optionale Unterstützung für cert-manager, eine weit verbreitete Lösung für die Verwaltung von TLS-Zertifikaten in Kubernetes, die den Prozess der Beschaffung, Verlängerung, Verwaltung und Verwendung dieser Zertifikate vereinfacht. Dies stellt sicher, dass Zertifikate gültig und aktuell sind, und versucht, Zertifikate zu einem konfigurierten Zeitpunkt vor Ablauf zu erneuern. cert-manager erleichtert auch die Ausstellung von Zertifikaten aus einer Vielzahl unterstützter Quellen, einschließlich AWS Certificate Manager Private Certificate Authority.

Wir empfehlen Ihnen, sich mit den bewährten Methoden für die Verwaltung von TLS-Zertifikaten auf Ihren Clustern vertraut zu machen und sich für Produktionsumgebungen für cert-manager zu entscheiden. Beachten Sie, dass Sie, wenn Sie sich für die Aktivierung von cert-manager für die Verwaltung der TLS-Zugangszertifikate von Webhook entscheiden, cert-manager auf Ihrem Amazon EKS-Cluster vorinstallieren müssen, bevor Sie das Amazon Observability EKS-Add-on installieren. CloudWatch Weitere Informationen zu den verfügbaren Installationsoptionen finden Sie in der cert-manager-Dokumentation. Nach der Installation können Sie sich bei der Erstellung oder Aktualisierung des Add-Ons dafür entscheiden, cert-manager für die Verwaltung der Webhook-TLS-Zugangszertifikate zu verwenden. Verwenden Sie dazu die folgende zusätzliche Konfiguration.

```
--configuration-values '{ "admissionWebhooks": { "certManager": { "enabled":
 true } } }'
```

Die in diesem Abschnitt beschriebene erweiterte Konfiguration verwendet standardmäßig einen Aussteller. SelfSigned

Erfassung von Amazon-EBS-Volume-IDs

Wenn Sie Amazon-EBS-Volume-IDs in den Leistungsprotokollen erfassen möchten, müssen Sie der IAM-Rolle, die den Worker-Knoten oder dem Servicekonto zugeordnet ist, eine weitere Richtlinie hinzufügen. Fügen Sie Folgendes als eingebundenen Richtlinie hinzu. Weitere Informationen finden Sie unter Hinzufügen und Entfernen von IAM-Identitätsberechtigungen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeVolumes"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Problembehandlung beim Amazon CloudWatch Observability EKS-Add-on

Verwenden Sie die folgenden Informationen, um Probleme mit dem Amazon CloudWatch Observability EKS-Add-on zu beheben.

Aktualisieren und Löschen des Amazon CloudWatch Observability EKS-Add-ons

Anweisungen zum Aktualisieren oder Löschen des Amazon CloudWatch Observability EKS-Add-ons finden Sie unter Amazon EKS-Add-Ons verwalten. Verwenden Sie als `amazon-cloudwatch-observability` Name des Add-Ons.

Überprüfen Sie die Version des CloudWatch Agenten, der vom Amazon CloudWatch Observability EKS-Add-on verwendet wird

Das Amazon CloudWatch Observability EKS-Add-on installiert eine benutzerdefinierte Ressource`AmazonCloudWatchAgent`, die das Verhalten des CloudWatch Agenten-Daemonsets auf dem Cluster steuert, einschließlich der Version des verwendeten CloudWatch Agenten. Sie können eine Liste aller auf Ihrem Cluster installierten, benutzerdefinierten `AmazonCloudWatchAgent`-Ressourcen abrufen, indem Sie den folgenden Befehl eingeben:

```
kubectl get amazoncloudwatchagent -A
```

In der Ausgabe dieses Befehls sollten Sie die Version des Agenten überprüfen können. CloudWatch Alternativ können Sie auch die `amazoncloudwatchagent`-Ressource oder einen der `cloudwatch-agent-*`-Pods beschreiben, die auf Ihrem Cluster ausgeführt werden, um das verwendete Image zu überprüfen.

Umgang mit a ConfigurationConflict bei der Verwaltung des Add-ons

Wenn Sie bei der Installation oder Aktualisierung des Amazon CloudWatch Observability EKS-Add-ons einen Fehler feststellen, `Health Issue` der durch einen Typ `ConfigurationConflict` mit einer Beschreibung verursacht wird, die mit beginnt`Conflicts found when trying to apply. Will not continue due to resolve conflicts mode`, liegt das wahrscheinlich daran, dass Sie den CloudWatch Agenten und die zugehörigen Komponenten wie den ServiceAccount, den ClusterRole und den bereits auf dem Cluster ClusterRoleBinding installiert haben. Wenn das Add-on versucht, den CloudWatch Agenten und die zugehörigen Komponenten zu installieren und eine Änderung des Inhalts feststellt, schlägt es standardmäßig die Installation oder Aktualisierung fehl, um zu verhindern, dass der Status der Ressourcen auf dem Cluster überschrieben wird.

Wenn Sie versuchen, das Amazon CloudWatch Observability EKS-Add-on zu integrieren und dieser Fehler auftritt, empfehlen wir, ein vorhandenes CloudWatch Agenten-Setup zu löschen, das Sie zuvor auf dem Cluster installiert hatten, und dann das EKS-Add-on zu installieren. Stellen Sie sicher, dass Sie alle Anpassungen, die Sie möglicherweise am ursprünglichen CloudWatch Agenten-Setup vorgenommen haben, wie z. B. eine benutzerdefinierte Agentenkonfiguration, sichern und diese dem Amazon CloudWatch Observability EMS-Add-on zur Verfügung stellen, wenn Sie es das nächste Mal installieren oder aktualisieren. Wenn Sie den CloudWatch Agenten für das Onboarding in Container Insights bereits installiert hatten, finden Sie Den CloudWatch Agenten und Fluent Bit für Container Insights löschen weitere Informationen unter.

Alternativ unterstützt das Add-On eine Konfigurationsoption zur Konfliktlösung, die `OVERWRITE` spezifizieren kann. Sie können diese Option verwenden, um mit der Installation oder Aktualisierung des Add-Ons fortzufahren, indem Sie die Konflikte auf dem Cluster überschreiben. Wenn Sie die Amazon-EKS-Konsole verwenden, finden Sie die Methode zur Konfliktlösung, wenn Sie beim Erstellen oder Aktualisieren des Add-Ons die optionalen Konfigurationseinstellungen auswählen. Wenn Sie den verwenden AWS CLI, können Sie den in Ihren Befehl eingeben, `--resolve-conflicts OVERWRITE` um das Add-on zu erstellen oder zu aktualisieren.

Vom CloudWatch Agenten gesammelte Metriken

Sie können Messwerte von Servern sammeln, indem Sie den CloudWatch Agenten auf dem Server installieren. Sie können den Agenten sowohl auf Amazon-EC2-Instances als auch On-Premises-Servern und auf Computern mit Linux, Windows Server oder macOS installieren. Wenn Sie den Agent auf einer Amazon-EC2-Instance installieren, werden die von ihm erfassten Metriken zusätzlich zu den Metriken erfasst, die auf Amazon-EC2-Instances standardmäßig aktiviert sind.

Informationen zur Installation des CloudWatch Agenten auf einer Instanz finden Sie unterErfassen Sie mit dem CloudWatch Agenten Metriken, Logs und Traces.

Alle in diesem Abschnitt besprochenen Metriken werden direkt vom CloudWatch Agenten erfasst.

Vom CloudWatch Agenten auf Windows Server-Instanzen gesammelte Metriken

Auf einem Server, auf dem Windows Server ausgeführt wird, können Sie durch die Installation des CloudWatch Agenten die Messwerte erfassen, die den Leistungsindikatoren im Windows-Leistungsmonitor zugeordnet sind. Die CloudWatch Messobjektnamen für diese Leistungsindikatoren werden erstellt, indem ein Leerzeichen zwischen dem Objektnamen und dem Leistungsindikatornamen eingefügt wird. Beispielsweise erhält der `% Interrupt Time` Zähler des `Processor` Objekts den Metriknamen `Processor % Interrupt Time` in CloudWatch. Weitere Informationen zu Leistungsindikatoren der Windows-Leistungsüberwachung finden Sie in der Dokumentation von Microsoft Windows Server.

Der Standard-Namespace für vom CloudWatch Agenten gesammelte Metriken ist`CWAgent`, obwohl Sie bei der Konfiguration des Agenten einen anderen Namespace angeben können.

Vom CloudWatch Agenten auf Linux- und macOS-Instances gesammelte Metriken

In der folgenden Tabelle sind die Metriken aufgeführt, die Sie mit dem CloudWatch Agenten auf Linux-Servern und macOS-Computern sammeln können.

Metrik	Beschreibung
`cpu_time_active`	Die Zeit, für die die CPU auf beliebige Art und Weise aktiv ist. Diese Metrik wird in Hundertstelsekunden gemessen. Einheit: keine
`cpu_time_guest`	Die Zeit, für die die CPU eine virtuelle CPU für ein Gastbetriebssystem zur Verfügung stellt. Diese Metrik wird in Hundertstelsekunden gemessen. Einheit: keine

Metrik	Beschreibung
cpu_time_guest_nice	Die Zeitspanne, in der die CPU eine virtuelle CPU für ein Gastbetriebssystem betreibt, die niedrige Priorität hat und durch andere Prozesse unterbrochen werden kann. Diese Metrik wird in Hundertstelsekunden gemessen. Einheit: keine
cpu_time_idle	Die Zeit, für die sich die CPU im Leerlauf befindet. Diese Metrik wird in Hundertstelsekunden gemessen. Einheit: keine
cpu_time_iowait	Die Zeit, für die die CPU auf I/O-Vorgänge wartet. Diese Metrik wird in Hundertstelsekunden gemessen. Einheit: keine
cpu_time_irq	Die Zeit, für die die CPU Unterbrechungen bedient. Diese Metrik wird in Hundertstelsekunden gemessen. Einheit: keine
cpu_time_nice	Die Zeitspanne, in der sich die CPU im Benutzerm odus mit Prozessen mit niedriger Priorität befindet, die leicht durch Prozesse mit höherer Priorität unterbrochen werden können. Diese Metrik wird in Hundertstelsekunden gemessen. Einheit: keine
cpu_time_softirq	Die Zeit, für die die CPU Softwareunterbrechungen bedient. Diese Metrik wird in Hundertstelsekunden gemessen. Einheit: keine

Metrik	Beschreibung
`cpu_time_steal`	Die Zeit, für die sich die CPU in gestohlener Zeit befindet. Dies ist die Zeit, die in anderen Betriebss ystemen in einer virtualisierten Umgebung verbracht wird. Diese Metrik wird in Hundertstelsekunden gemessen. Einheit: keine
`cpu_time_system`	Die Zeit, die die CPU im Systemmodus verbringt. Diese Metrik wird in Hundertstelsekunden gemessen. Einheit: keine
`cpu_time_user`	Die Zeit, die die CPU im Benutzermodus verbringt. Diese Metrik wird in Hundertstelsekunden gemessen. Einheit: keine
`cpu_usage_active`	Der Prozentsatz der Zeit, für die die CPU auf beliebige Art und Weise aktiv ist. Einheit: Prozent
`cpu_usage_guest`	Der Prozentanteil der Zeit, für die die CPU eine virtuelle CPU für ein Gastbetriebssystem zur Verfügung stellt. Einheit: Prozent
`cpu_usage_guest_nice`	Der Prozentsatz der Zeit, in der die CPU eine virtuelle CPU für ein Gastbetriebssystem betreibt, der niedrige Priorität hat und durch andere Prozesse unterbrochen werden kann. Einheit: Prozent

Metrik	Beschreibung
`cpu_usage_idle`	Der Prozentsatz der Zeit, die sich die CPU im Leerlauf befindet. Einheit: Prozent
`cpu_usage_iowait`	Der Prozentanteil der Zeit, für die die CPU auf I/O-Vorgänge wartet. Einheit: Prozent
`cpu_usage_irq`	Der Prozentanteil der Zeit, für die die CPU Unterbrechungen bedient. Einheit: Prozent
`cpu_usage_nice`	Der Anteil der Zeit, in der sich die CPU im Benutzermodus mit Prozessen mit niedriger Priorität befindet, die leicht durch Prozesse mit höherer Priorität unterbrochen werden können. Einheit: Prozent
`cpu_usage_softirq`	Der Prozentanteil der Zeit, für die die CPU Softwareunterbrechungen bedient. Einheit: Prozent
`cpu_usage_steal`	Der Anteil der Zeit, für den sich die CPU in gestohlener Zeit oder Zeit, die in anderen Betriebssystemen in einer virtualisierten Umgebung verbracht wird, befindet. Einheit: Prozent
`cpu_usage_system`	Der Prozentanteil der Zeit, die die CPU im Systemmodus verbringt. Einheit: Prozent

Metrik	Beschreibung
cpu_usage_user	Der Prozentanteil der Zeit, die die CPU im Benutzermodus verbringt. Einheit: Prozent
disk_free	Freier Speicherplatz auf den Festplatten. Einheit: Byte
disk_inodes_free	Die Anzahl der verfügbaren Index-Knoten auf der Festplatte. Einheit: Anzahl
disk_inodes_total	Die Gesamtanzahl der reservierten Index-Knoten auf der Festplatte. Einheit: Anzahl
disk_inodes_used	Die Anzahl der verwendeten Index-Knoten auf der Festplatte. Einheit: Anzahl
disk_total	Der Gesamtspeicherplatz auf den Festplatten, sowohl verwendet als auch frei. Einheit: Byte
disk_used	Verwendeter Speicherplatz auf den Festplatten. Einheit: Byte
disk_used_percent	Der Prozentanteil des verwendeten Gesamtspe icherplatzes. Einheit: Prozent

Metrik	Beschreibung
`diskio_iops_in_progress`	Die Anzahl von I/O-Anforderungen, die an den Gerätetreiber gestellt wurden, das Gerät jedoch noch nicht abgeschlossen hat. Einheit: Anzahl
`diskio_io_time`	Die Zeit, für die sich I/O-Anforderungen in der Warteschlange des Datenträgers befinden. Einheit: Millisekunden Die einzige Statistik, die für diese Metrik verwendet werden sollte, ist `Sum`. Verwenden Sie nicht `Average`.
`diskio_reads`	Die Anzahl der Festplattenlesevorgänge. Einheit: Anzahl Die einzige Statistik, die für diese Metrik verwendet werden sollte, ist `Sum`. Verwenden Sie nicht `Average`.
`diskio_read_bytes`	Die Anzahl der von den Festplatten gelesenen Bytes. Einheit: Byte Die einzige Statistik, die für diese Metrik verwendet werden sollte, ist `Sum`. Verwenden Sie nicht `Average`.

Metrik	Beschreibung
`diskio_read_time`	Die Zeit, für die Leseanforderungen auf den Festplatten gewartet haben. Mehrere gleichzeitig wartende Leseanforderungen erhöhen die Anzahl. Wenn beispielsweise 5 Anfragen im Mittel 100 Millisekunden lang warten, wird 500 gemeldet. Einheit: Millisekunden Die einzige Statistik, die für diese Metrik verwendet werden sollte, ist `Sum`. Verwenden Sie nicht `Average`.
`diskio_writes`	Die Anzahl der Festplattenschreibvorgänge. Einheit: Anzahl Die einzige Statistik, die für diese Metrik verwendet werden sollte, ist `Sum`. Verwenden Sie nicht `Average`.
`diskio_write_bytes`	Anzahl der auf die Festplatten geschriebenen Bytes. Einheit: Byte Die einzige Statistik, die für diese Metrik verwendet werden sollte, ist `Sum`. Verwenden Sie nicht `Average`.

Metrik	Beschreibung
`diskio_write_time`	Die Zeit, für die Schreibanforderungen auf den Festplatten gewartet haben. Mehrere gleichzeitig wartende Schreibanforderungen erhöhen die Anzahl. Wenn beispielsweise 8 Anfragen im Mittel 1000 Millisekunden lang warten, wird 8000 gemeldet. Einheit: Millisekunden Die einzige Statistik, die für diese Metrik verwendet werden sollte, ist `Sum`. Verwenden Sie nicht `Average`.
`ethtool_bw_in_allowance_exceeded`	Die Anzahl der Pakete, die in die Warteschlange gestellt und/oder verworfen wurden, da die eingehende aggregierte Bandbreite das Maximum für die Instance überschritten hat. Diese Metrik wird nur erfasst, wenn Sie sie im `ethtool` Unterabschnitt des `metrics_collected` Abschnitts der CloudWatch Agenten-Konfigurationsdatei aufgeführt haben. Weitere Informationen finden Sie unter Netzwerkleistungsmetriken sammeln. Einheit: keine

Metrik	Beschreibung
`ethtool_bw_out_allowance_exceeded`	Die Anzahl der Pakete, die in die Warteschlange gestellt und/oder verworfen wurden, weil die ausgehende aggregierte Bandbreite das Maximum für die Instance überschritten hat. Diese Metrik wird nur erfasst, wenn Sie sie im `ethtool` Unterabschnitt des `metrics_collected` Abschnitts der CloudWatch Agenten-Konfigurationsdatei aufgeführt haben. Weitere Informationen finden Sie unter Netzwerkleistungsmetriken sammeln. Einheit: keine
`ethtool_conntrack_allowance_exceeded`	Die Anzahl der verworfenen Pakete, weil die Verbindungsverfolgung das Maximum für die Instance überschritten hat und keine neuen Verbindungen hergestellt werden konnten. Dies kann zu einem Paketverlust für den Datenverkehr zur oder von der Instance führen. Diese Metrik wird nur erfasst, wenn Sie sie im `ethtool` Unterabschnitt des `metrics_collected` Abschnitts der CloudWatch Agenten-Konfigurationsdatei aufgeführt haben. Weitere Informationen finden Sie unter Netzwerkleistungsmetriken sammeln. Einheit: keine

Metrik	Beschreibung
`ethtool_linklocal_allowance_exceeded`	Die Anzahl der verworfenen Pakete, weil das PPS des Datenverkehrs zu lokalen Proxy-Diensten das Maximum für die Netzwerkschnittstelle überschritten hat. Dies wirkt sich auf den Datenverkehr zum DNS-Dienst, zum Instance Metadata Service und zum Amazon Time Sync Service aus. Diese Metrik wird nur erfasst, wenn Sie sie im `ethtool` Unterabschnitt des `metrics_collected` Abschnitts der CloudWatch Agenten-Konfigurationsdatei aufgeführt haben. Weitere Informationen finden Sie unter Netzwerkleistungsmetriken sammeln. Einheit: keine
`ethtool_pps_allowance_exceeded`	Die Anzahl der Pakete, die in die Warteschlange gestellt und/oder verworfen wurden, weil die bidirektionale PPS das Maximum für die Instance überschritten hat. Diese Metrik wird nur erfasst, wenn Sie sie im `ethtool` Unterabschnitt des `metrics_collected` Abschnitts der CloudWatch Agenten-Konfigurationsdatei aufgeführt haben. Weitere Informationen finden Sie unter Netzwerkleistungsmetriken sammeln. Einheit: keine
`mem_active`	Die Speichermenge, die während des letzten Stichprobenzeitraums auf beliebige Art und Weise verwendet wurde. Einheit: Byte

Metrik	Beschreibung
mem_available	Die Speichermenge, die verfügbar ist und sofort Prozessen zugewiesen werden kann. Einheit: Byte
mem_available_percent	Der Prozentanteil des Speichers, der verfügbar ist und sofort Prozessen zugewiesen werden kann. Einheit: Prozent
mem_buffered	Die Speichermenge, die für Puffer verwendet wird. Einheit: Byte
mem_cached	Die Speichermenge, die für Datei-Caches verwendet wird. Einheit: Byte
mem_free	Die Speichermenge, die nicht verwendet wird. Einheit: Byte
mem_inactive	Die Speichermenge, die während des letzten Stichprobenzeitraums nicht verwendet wurde. Einheit: Byte
mem_total	Die Gesamtgröße des Speichers. Einheit: Byte
mem_used	Die derzeit verwendete Speichermenge. Einheit: Byte
mem_used_percent	Der derzeit verwendete Anteil des Speicherplatzes in Prozent. Einheit: Prozent

Metrik	Beschreibung
net_bytes_recv	Die Anzahl der von der Netzwerkschnittstelle empfangenen Bytes. Einheit: Byte Die einzige Statistik, die für diese Metrik verwendet werden sollte, ist Sum. Verwenden Sie nicht Average.
net_bytes_sent	Die Anzahl der von der Netzwerkschnittstelle gesendeten Bytes. Einheit: Byte Die einzige Statistik, die für diese Metrik verwendet werden sollte, ist Sum. Verwenden Sie nicht Average.
net_drop_in	Die Anzahl der von dieser Netzwerkschnittstelle empfangenen Pakete, die gelöscht wurden. Einheit: Anzahl Die einzige Statistik, die für diese Metrik verwendet werden sollte, ist Sum. Verwenden Sie nicht Average.
net_drop_out	Die Anzahl der von dieser Netzwerkschnittstelle übertragenen Pakete, die gelöscht wurden. Einheit: Anzahl Die einzige Statistik, die für diese Metrik verwendet werden sollte, ist Sum. Verwenden Sie nicht Average.

Metrik	Beschreibung
net_err_in	Die Anzahl der Empfangsfehler, die diese Netzwerks chnittstelle erkannt hat. Einheit: Anzahl Die einzige Statistik, die für diese Metrik verwendet werden sollte, ist Sum. Verwenden Sie nicht Average.
net_err_out	Die Anzahl der Übertragungsfehler, die diese Netzwerkschnittstelle erkannt hat. Einheit: Anzahl Die einzige Statistik, die für diese Metrik verwendet werden sollte, ist Sum. Verwenden Sie nicht Average.
net_packets_sent	Die Anzahl der von dieser Netzwerkschnittstelle gesendeten Pakete. Einheit: Anzahl Die einzige Statistik, die für diese Metrik verwendet werden sollte, ist Sum. Verwenden Sie nicht Average.
net_packets_recv	Die Anzahl der von dieser Netzwerkschnittstelle empfangenen Pakete. Einheit: Anzahl Die einzige Statistik, die für diese Metrik verwendet werden sollte, ist Sum. Verwenden Sie nicht Average.

Metrik	Beschreibung
`netstat_tcp_close`	Die Anzahl der TCP-Verbindungen ohne Status. Einheit: Anzahl
`netstat_tcp_close_wait`	Die Anzahl der TCP-Verbindungen, die auf eine Beendigungsanforderung vom Client warten. Einheit: Anzahl
`netstat_tcp_closing`	Die Anzahl der TCP-Verbindungen, die auf eine Beendigungsanforderung mit Bestätigung vom Client warten. Einheit: Anzahl
`netstat_tcp_established`	Die Anzahl der eingerichteten TCP-Verbindungen. Einheit: Anzahl
`netstat_tcp_fin_wait1`	Die Anzahl der TCP-Verbindungen mit Status `FIN_WAIT1` während des Schließens einer Verbindung. Einheit: Anzahl
`netstat_tcp_fin_wait2`	Die Anzahl der TCP-Verbindungen mit Status `FIN_WAIT2` während des Schließens einer Verbindung. Einheit: Anzahl
`netstat_tcp_last_ack`	Die Anzahl der TCP-Verbindungen, die darauf warten, dass der Client die Nachricht über die Beendigung der Verbindung bestätigt. Dies ist der letzte Status, bevor die Verbindung geschlossen wird. Einheit: Anzahl

Metrik	Beschreibung
`netstat_tcp_listen`	Die Anzahl der TCP-Ports, die derzeit auf eine Verbindung warten. Einheit: Anzahl
`netstat_tcp_none`	Die Anzahl der TCP-Verbindungen mit inaktiven Clients. Einheit: Anzahl
`netstat_tcp_syn_sent`	Die Anzahl der TCP-Verbindungen, die nach dem Senden einer Verbindungsanforderung auf eine übereinstimmende Verbindungsanforderung warten. Einheit: Anzahl
`netstat_tcp_syn_recv`	Die Anzahl der TCP-Verbindungen, die nach dem Senden und Empfangen einer Verbindungsanforde rung auf eine Anforderungsbestätigung warten. Einheit: Anzahl
`netstat_tcp_time_wait`	Die Anzahl der TCP-Verbindungen, die derzeit darauf warten, dass der Client die Bestätigung seiner Verbindungsabbauanforderung erhält. Einheit: Anzahl
`netstat_udp_socket`	Die Anzahl der aktuellen UDP-Verbindungen. Einheit: Anzahl
`processes_blocked`	Die Anzahl von blockierten Prozessen. Einheit: Anzahl

Metrik	Beschreibung
processes_dead	Die Anzahl der „toten“ Prozesse, die unter Linux den Statuscode X tragen. Diese Metrik wird auf macOS-Computern nicht erfasst. Einheit: Anzahl
processes_idle	Anzahl der Prozesse, die sich im Leerlauf befinden, für die also länger als 20 Sekunden keine Aktivität en stattgefunden hat. Nur auf FreeBSD-Instances verfügbar. Einheit: Anzahl
processes_paging	Die Anzahl der ausgelagerten Prozesse, die unter Linux den Statuscode W tragen. Diese Metrik wird auf macOS-Computern nicht erfasst. Einheit: Anzahl
processes_running	Die Anzahl der laufenden Prozesse, angezeigt durch den Statuscode R. Einheit: Anzahl
processes_sleeping	Die Anzahl der Prozesse im Standby-Modus, angezeigt durch den Statuscode S. Einheit: Anzahl
processes_stopped	Die Anzahl der angehaltenen Prozesse, angezeigt durch den Statuscode T. Einheit: Anzahl

Metrik	Beschreibung
`processes_total`	Die Gesamtanzahl der Prozesse auf der Instance. Einheit: Anzahl
`processes_total_threads`	Die Gesamtanzahl der Threads der Prozesse. Diese Metrik ist nur für Linux-Instances verfügbar. Diese Metrik wird auf macOS-Computern nicht erfasst. Einheit: Anzahl
`processes_wait`	Die Anzahl der ausgelagerten Prozesse, die in FreeBSD-Instances den Statuscode `W` tragen. Diese Metrik ist nur auf FreeBSD-Instances und nicht auf Linux-, Windows Server- oder macOS-Instances verfügbar. Einheit: Anzahl
`processes_zombies`	Die Anzahl der Zombieprozesse, angezeigt durch Statuscode `Z`. Einheit: Anzahl
`swap_free`	Der Speicherplatz des Auslagerungsbereichs, der nicht verwendet wird. Einheit: Byte
`swap_used`	Der Speicherplatz des Auslagerungsbereichs, der derzeit verwendet wird. Einheit: Byte
`swap_used_percent`	Der Prozentanteil des Auslagerungsbereichs, der derzeit verwendet wird. Einheit: Prozent

Definitionen der vom Agenten gesammelten Speichermetriken CloudWatch

Wenn der CloudWatch Agent Speichermetriken sammelt, ist die Quelle das Speicherverwaltungs-Subsystem des Hosts. Zum Beispiel legt der Linux-Kernel die vom Betriebssystem verwalteten Daten in `/proc` offen. Was den Arbeitsspeicher betrifft, so befinden sich die Daten in `/proc/meminfo`.

Für jedes Betriebssystem und jede Architektur werden die Ressourcen, die von Prozessen verwendet werden, unterschiedlich berechnet. Weitere Informationen finden Sie in den folgenden Abschnitten.

Während jedes Erfassungsintervalls sammelt der CloudWatch Agent auf jeder Instanz die Instanzressourcen und berechnet die Ressourcen, die von allen Prozessen verwendet werden, die in dieser Instanz ausgeführt werden. Diese Informationen werden in CloudWatch Metriken zurückgemeldet. Sie können die Länge des Erfassungsintervalls in der CloudWatch Agent-Konfigurationsdatei konfigurieren. Weitere Informationen finden Sie unter CloudWatch Agenten-Konfigurationsdatei: Abschnitt Agent.

In der folgenden Liste wird erklärt, wie die Speichermetriken definiert sind, die der CloudWatch Agent erfasst.

- Aktiver Speicher – Speicher, der von einem Prozess verwendet wird. Mit anderen Worten, der Speicher, der von aktuell laufenden Anwendungen verwendet wird.
- Verfügbarer Speicher – Der Speicher, der den Prozessen sofort zur Verfügung gestellt werden kann, ohne dass das System in den Swap wechselt (auch als virtueller Speicher bezeichnet).
- Pufferspeicher – Der Datenbereich, der von Hardwaregeräten oder Programmprozessen gemeinsam genutzt wird, die mit unterschiedlichen Geschwindigkeiten und Prioritäten arbeiten.
- Zwischenspeicher – Speichert Programmanweisungen und Daten, die wiederholt bei der Ausführung von Programmen verwendet werden, die die CPU wahrscheinlich als Nächstes benötigt.
- Freier Speicher – Speicher, der überhaupt nicht verwendet wird und sofort verfügbar ist. Es ist völlig kostenlos, dass das System bei Bedarf verwendet werden kann.
- Inaktiver Speicher – Seiten, auf die „kürzlich“ nicht zugegriffen wurde.
- Gesamtspeicher – Die Größe des tatsächlichen physischen RAM-Speichers.
- Verwendeter Speicher – Speicher, der derzeit von Programmen und Prozessen verwendet wird.

Themen

- macOS: Gesammelte Metriken und verwendete Berechnungen
- Windows: Gesammelte Metriken
- Beispiel: Berechnung von Speichermetriken auf Linux

Linux: Gesammelte Metriken und verwendete Berechnungen

Gesammelte Metriken und Einheiten:

- Aktiv (Byte)
- Verfügbar (Byte)
- Verfügbarer Prozentsatz (Prozent)
- Gepuffert (Byte)
- Zwischengespeichert (Byte)
- Kostenlos (Byte)
- Inaktiv (Byte)
- Gesamt (Byte)
- Benutzt (Byte)
- Verwendeter Prozentsatz (Prozent)

Verwendeter Speicher = Gesamtspeicher - Freier Speicher - Zwischenspeicher - Pufferspeicher

Gesamtspeicher = Verwendeter Speicher + Freier Speicher + Zwischenspeicher + Pufferspeicher

macOS: Gesammelte Metriken und verwendete Berechnungen

Gesammelte Metriken und Einheiten:

- Aktiv (Byte)
- Verfügbar (Byte)
- Verfügbarer Prozentsatz (Prozent)
- Kostenlos (Byte)
- Inaktiv (Byte)
- Gesamt (Byte)
- Benutzt (Byte)

- Verwendeter Prozentsatz (Prozent)

Verfügbarer Speicher = Freier Speicher + Inaktiver Speicher

Verwendeter Speicher = Gesamtspeicher - Verfügbarer Speicher

Gesamtspeicher = Verfügbarer Speicher + Verwendeter Speicher

Windows: Gesammelte Metriken

Die auf Windows-Hosts erfassten Metriken sind unten aufgeführt. Alle diese Metriken haben `None` für `Unit`.

- Verfügbare Byte
- Cache-Fehler/Sekunde
- Seitenfehler/Sekunde
- Seiten/Sekunde

Für Windows-Metriken werden keine Berechnungen verwendet, da der CloudWatch Agent Ereignisse anhand von Leistungsindikatoren analysiert.

Beispiel: Berechnung von Speichermetriken auf Linux

Nehmen wir als Beispiel an, dass die Eingabe des **`cat /proc/meminfo`**-Befehls auf einem Linux-Host zu folgenden Ergebnissen führt:

```
MemTotal:       3824388 kB
MemFree:         462704 kB
MemAvailable:   2157328 kB
Buffers:         126268 kB
Cached:         1560520 kB
SReclaimable:    289080 kB>
```

In diesem Beispiel erfasst der CloudWatch Agent die folgenden Werte. Alle Werte, die der CloudWatch Agent sammelt und meldet, sind in Byte angegeben.

- `mem_total`: 3916173312 Byte
- `mem_available`: 2209103872 Byte (+ zwischengespeichert) MemFree

- `mem_free`: 473808896 Byte
- `mem_cached`: 1893990400 Byte (`cached` + `SReclaimable`
- `mem_used`: 1419075584 Byte (`MemTotal` – (`MemFree` + `Buffers` + (`Cached` + `SReclaimable`)))
- `mem_buffered`: 129667072 Byte
- `mem_available_percent`: 56,41 %
- `mem_used_percent`: 36,24 % (`mem_used`/`mem_total`) * 100

Häufige Szenarien mit dem Agenten CloudWatch

In den folgenden Abschnitten wird beschrieben, wie allgemeine Konfigurations- und Anpassungsaufgaben für den CloudWatch Agenten ausgeführt werden.

Themen

Den CloudWatch Agenten unter einem anderen Benutzer ausführen

Auf Linux-Servern CloudWatch wird der standardmäßig als Root-Benutzer ausgeführt. Damit der Agent unter einem anderen Benutzer ausgeführt wird, verwenden Sie den `run_as_user` Parameter im `agent` Abschnitt in der CloudWatch Agenten-Konfigurationsdatei. Diese Option ist nur auf Linux-Servern verfügbar.

Wenn Sie den Agenten bereits mit dem Root-Benutzer ausführen und zu einem anderen Benutzer wechseln möchten, gehen Sie wie folgt vor.

Um den CloudWatch Agenten unter einem anderen Benutzer auf einer EC2-Instance unter Linux auszuführen

1. Laden Sie ein neues CloudWatch Agentenpaket herunter und installieren Sie es. Weitere Informationen finden Sie unter Laden Sie das CloudWatch Agentenpaket herunter.
2. Erstellen Sie einen neuen Linux-Benutzer oder verwenden Sie den Standardbenutzer `cwagent`, der die RPM- oder DEB-Datei erstellt hat.
3. Geben Sie auf eine der folgenden Arten Anmeldeinformationen für diesen Benutzer an:
   - Wenn die Datei im Home-Verzeichnis des Root-Benutzers `.aws/credentials` vorhanden ist, müssen Sie eine Anmeldeinformationsdatei für den Benutzer erstellen, mit dem Sie den CloudWatch Agenten ausführen möchten. Diese Datei mit den Anmeldeinformationen lautet `/home/`*`username`*`/.aws/credentials`. Setzen Sie dann den Wert des Parameters `shared_credential_file` in `common-config.toml` auf den Pfadnamen der Anmeldedatei. Weitere Informationen finden Sie unter (Optional) Ändern der gemeinsamen Konfiguration für Proxy- oder Regionsangaben.
   - Wenn die Datei `.aws/credentials` nicht im Heim-Verzeichnis des Stammbenutzers vorhanden ist, können Sie einen der folgenden Schritte ausführen:
     - Erstellen Sie eine Anmeldeinformationsdatei für den Benutzer, den Sie zum Ausführen des CloudWatch Agenten verwenden werden. Diese Datei mit den Anmeldeinformationen lautet `/home/`*`username`*`/.aws/credentials`. Setzen Sie dann den Wert des Parameters `shared_credential_file` in `common-config.toml` auf den Pfadnamen der Anmeldedatei. Weitere Informationen finden Sie unter (Optional) Ändern der gemeinsamen Konfiguration für Proxy- oder Regionsangaben.
     - Anstatt eine Datei mit Anmeldeinformationen zu erstellen, fügen Sie der Instance eine IAM-Rolle hinzu. Der Agent verwendet diese Rolle als Anmeldeprovider.
4. Fügen Sie in der CloudWatch Agent-Konfigurationsdatei im `agent` Abschnitt die folgende Zeile hinzu:

   ```
   "run_as_user": "username"
   ```

   Nehmen Sie bei Bedarf weitere Änderungen an der Konfigurationsdatei vor. Weitere Informationen finden Sie unter Erstellen Sie die CloudWatch Agent-Konfigurationsdatei.

5. Erteilen Sie dem Benutzer die erforderlichen Berechtigungen. Der Benutzer muss über Lese-(r)-Berechtigungen für die zu erfassenden Protokolldateien und über Execute-(x)-Berechtigung für jedes Verzeichnis im Pfad der Protokolldateien verfügen.
6. Starten Sie den Agenten mit der Konfigurationsdatei, die Sie gerade geändert haben.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -s -c file:configuration-file-path
```

Um den CloudWatch Agenten unter einem anderen Benutzer auf einem lokalen Server unter Linux auszuführen

1. Laden Sie ein neues CloudWatch Agentenpaket herunter und installieren Sie es. Weitere Informationen finden Sie unter Laden Sie das CloudWatch Agentenpaket herunter.
2. Erstellen Sie einen neuen Linux-Benutzer oder verwenden Sie den Standardbenutzer `cwagent`, der die RPM- oder DEB-Datei erstellt hat.
3. Speichern Sie die Anmeldeinformationen dieses Benutzers in einem Pfad, auf den der Benutzer zugreifen kann, z. B. `/home/username/.aws/credentials`.
4. Setzen Sie den Wert des Parameters `shared_credential_file` in `common-config.toml` auf den Pfadnamen der Anmeldedatei. Weitere Informationen finden Sie unter (Optional) Ändern der gemeinsamen Konfiguration für Proxy- oder Regionsangaben.
5. Fügen Sie in der CloudWatch Agent-Konfigurationsdatei die folgende Zeile in den `agent` Abschnitt ein:

   ```
   "run_as_user": "username"
   ```

   Nehmen Sie bei Bedarf weitere Änderungen an der Konfigurationsdatei vor. Weitere Informationen finden Sie unter Erstellen Sie die CloudWatch Agent-Konfigurationsdatei.
6. Erteilen Sie dem Benutzer erforderliche Berechtigungen. Der Benutzer muss über Lese-(r)-Berechtigungen für die zu erfassenden Protokolldateien und über Execute-(x)-Berechtigung für jedes Verzeichnis im Pfad der Protokolldateien verfügen.
7. Starten Sie den Agenten mit der Konfigurationsdatei, die Sie gerade geändert haben.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -s -c file:configuration-file-path
```

Wie der CloudWatch Agent mit spärlichen Protokolldateien umgeht

Sparse-Dateien sind Dateien mit leeren Blöcken und echten Inhalten. Eine Sparse-Datei verwendet Speicherplatz effizienter, indem anstelle der tatsächlichen Null-Bytes, aus denen der Block besteht, kurze Informationen, die die leeren Blöcke darstellen, auf die Festplatte geschrieben werden. Dadurch wird die tatsächliche Größe einer Sparse-Datei in der Regel viel kleiner als die scheinbare Größe.

Der CloudWatch Agent behandelt Dateien mit geringer Dichte jedoch nicht anders als normale Dateien. Wenn der Agent eine Sparse-Datei liest, werden die leeren Blöcke als „echte“ Blöcke behandelt, die mit Null-Bytes gefüllt sind. Aus diesem Grund veröffentlicht der CloudWatch Agent so viele Byte, wie die scheinbare Größe einer Datei mit geringer Dichte entspricht. CloudWatch

Wenn der CloudWatch Agent so konfiguriert wird, dass er eine Datei mit geringer Dichte veröffentlicht, kann dies zu höheren CloudWatch Kosten als erwartet führen. Wir empfehlen daher, dies nicht zu tun. Beispielsweise handelt es `/var/logs/lastlog` sich unter Linux normalerweise um eine Datei mit sehr geringer Dichte, und wir empfehlen, sie nicht in zu veröffentlichen. CloudWatch

Hinzufügen benutzerdefinierter Dimensionen zu den vom Agenten gesammelten Metriken CloudWatch

Um benutzerdefinierte Dimensionen wie Tags zu Metriken hinzuzufügen, die vom Agent erfasst werden, fügen Sie das Feld `append_dimensions` dem Abschnitt der Agent-Konfigurationsdatei hinzu, in dem diese Metriken aufgelistet sind.

Beispiel: Der folgende Beispielabschnitt der Konfigurationsdatei fügt eine benutzerdefinierte Dimension mit dem Namen `stackName` und dem Wert `Prod` zu den vom Agenten gesammelten Metriken `cpu` und `disk` hinzu.

```
"cpu":{
  "resources":[
    "*"
  ],
  "measurement":[
    "cpu_usage_guest",
    "cpu_usage_nice",
    "cpu_usage_idle"
  ],
```

```
  "totalcpu":false,
  "append_dimensions":{
    "stackName":"Prod"
  }
},
"disk":{
  "resources":[
    "/",
    "/tmp"
  ],
  "measurement":[
    "total",
    "used"
  ],
  "append_dimensions":{
    "stackName":"Prod"
  }
}
```

Beachten Sie, dass Sie den Agenten bei jeder Änderung der Agentenkonfigurationsdatei neu starten müssen, damit die Änderungen wirksam werden.

Mehrere CloudWatch Agenten-Konfigurationsdateien

Sowohl auf Linux- als auch auf Windows-Servern können Sie den CloudWatch Agenten so einrichten, dass er mehrere Konfigurationsdateien verwendet. Sie können zum Beispiel eine gemeinsame Konfigurationsdatei verwenden, die eine Reihe von Metriken, Protokollen und Ablaufverfolgungen sammelt, die Sie immer von allen Servern in Ihrer Infrastruktur sammeln möchten. Anschließend können Sie zusätzliche Konfigurationsdateien verwenden, die Metriken aus bestimmten Anwendungen oder in bestimmten Situationen erfassen.

Um dies einzurichten, erstellen Sie zunächst die Konfigurationsdateien, die Sie verwenden möchten. Alle Konfigurationsdateien, die gemeinsam auf demselben Server benutzt werden, müssen unterschiedliche Dateinamen haben. Sie können die Konfigurationsdateien auf Servern oder in Parameter Store speichern.

Starten Sie den CloudWatch Agenten mit der `fetch-config` Option und geben Sie die erste Konfigurationsdatei an. Um die zweite Konfigurationsdatei an den ausgeführten Agent anzufügen, verwenden Sie denselben Befehl, aber mit der `append-config`-Option. Alle Metriken, Protokolle und Ablaufverfolgungen, die in einer der beiden Konfigurationsdateien aufgeführt sind, werden gesammelt. Die folgenden Beispielbefehle veranschaulichen dieses Szenario mithilfe von

Konfigurationen, die als Dateien gespeichert sind. Die erste Zeile startet den Agent mithilfe der `infrastructure.json`-Konfigurationsdatei und die zweite Zeile fügt die `app.json`-Konfigurationsdatei an.

Die folgenden Beispielbefehle gelten für Linux.

```
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2
 -s -c file:/tmp/infrastructure.json
```

```
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a append-config -m
 ec2 -s -c file:/tmp/app.json
```

Die folgenden Beispielbefehle gelten für Windows Server.

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
 -a fetch-config -m ec2 -s -c file:"C:\Program Files\Amazon\AmazonCloudWatchAgent
\infrastructure.json"
```

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
 -a append-config -m ec2 -s -c file:"C:\Program Files\Amazon\AmazonCloudWatchAgent
\app.json"
```

Die folgenden Beispiel-Konfigurationsdateien veranschaulichen eine Nutzung für dieses Feature. Die erste Konfigurationsdatei wird für alle Server in der Infrastruktur verwendet, und die zweite erfasst nur Protokolle von einer bestimmten Anwendung und wird an Server angehängt, die die Anwendung ausführen.

infrastructure.json

```
{
  "metrics": {
    "metrics_collected": {
      "cpu": {
        "resources": [
          "*"
        ],
        "measurement": [
          "usage_active"
        ],
        "totalcpu": true
```

```
      },
      "mem": {
         "measurement": [
           "used_percent"
        ]
      }
    }
  },
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-
agent.log",
            "log_group_name": "amazon-cloudwatch-agent.log"
          },
          {
            "file_path": "/var/log/messages",
            "log_group_name": "/var/log/messages"
          }
        ]
      }
    }
  }
}
```

app.json

```
{
    "logs": {
        "logs_collected": {
            "files": {
                "collect_list": [
                    {
                        "file_path": "/app/app.log*",
                        "log_group_name": "/app/app.log"
                    }
                ]
            }
        }
    }
}
```

Die Dateinamen aller Konfigurationsdateien, die an die Konfiguration angehängt werden, müssen sich voneinander und vom Namen der anfänglichen Konfigurationsdatei unterscheiden. Wenn Sie `append-config` mit einer Konfigurationsdatei mit dem selben Dateinamen wie eine Konfigurationsdatei verwenden, die der Agent bereits verwendet, überschreibt der Append-Befehl die Informationen aus der ersten Konfigurationsdatei, anstatt Inhalte anzuhängen. Dies gilt auch, wenn sich zwei Konfigurationsdateien mit demselben Dateinamen in verschiedenen Dateipfaden befinden.

Das vorstehende Beispiel zeigt die Verwendung von zwei Konfigurationsdateien. Es gibt jedoch keine Beschränkungen hinsichtlich der Anzahl der Konfigurationsdateien, die Sie an die Agentenkonfiguration anhängen können. Sie können auch die Nutzung von Konfigurationsdateien auf Servern und Konfigurationen in Parameter Store kombinieren.

Aggregation oder Zusammenfassung der vom Agenten gesammelten Metriken CloudWatch

Um vom Agenten erfasste Metriken zu aggregieren oder zusammenzufassen, fügen Sie im Abschnitt für die betreffende Metrik in der Agentenkonfigurationsdatei das Feld `aggregation_dimensions` hinzu.

Beispielsweise fasst der folgende Abschnitt der Konfigurationsdatei Metriken in der `AutoScalingGroupName`-Dimension zusammen. Die Metriken aus allen Instances in der jeweiligen Auto-Scaling-Gruppe werden aggregiert und können als Ganzes angezeigt werden.

```
"metrics": {
  "cpu":{...}
  "disk":{...}
  "aggregation_dimensions" : [["AutoScalingGroupName"]]
}
```

Wenn Sie zusätzlich zum Zusammenfassen im Auto-Scaling-Gruppennamen auch die Kombination der einzelnen `InstanceId`- und `InstanceType`-Dimensionen zusammenfassen möchten, können Sie Folgendes hinzufügen.

```
"metrics": {
  "cpu":{...}
  "disk":{...}
  "aggregation_dimensions" : [["AutoScalingGroupName"], ["InstanceId", "InstanceType"]]
}
```

Wenn Sie stattdessen Metriken in einer Sammlung zusammenfassen möchten, verwenden Sie `[]`.

```
"metrics": {
  "cpu":{...}
  "disk":{...}
  "aggregation_dimensions" : [[]]
}
```

Beachten Sie, dass Sie den Agenten bei jeder Änderung der Agentenkonfigurationsdatei neu starten müssen, damit die Änderungen wirksam werden.

Erfassung hochauflösender Metriken mit dem Agenten CloudWatch

Das `metrics_collection_interval`-Feld gibt das Zeitintervall für die erfassten Metriken in Sekunden an. Durch Angabe eines Werts von weniger als 60 für dieses Feld werden die Metriken als hochauflösende Metriken erfasst.

Beispiel: Wenn Sie möchten, dass alle Ihre Metriken hochauflösend sind und alle 10 Sekunden erfasst werden, geben Sie als Wert für `metrics_collection_interval` im Abschnitt `agent` „10“ als Intervall für die globale Erfassung von Metriken an.

```
"agent": {
  "metrics_collection_interval": 10
}
```

Alternativ wird im folgenden Beispiel festgelegt, dass die `cpu`-Metriken jede Sekunde erfasst werden, während andere Metriken jede Minute erfasst werden.

```
"agent":{
  "metrics_collection_interval": 60
},
"metrics":{
  "metrics_collected":{
    "cpu":{
      "resources":[
        "*"
      ],
      "measurement":[
        "cpu_usage_guest"
      ],
```

```
      "totalcpu":false,
      "metrics_collection_interval": 1
    },
    "disk":{
      "resources":[
        "/",
        "/tmp"
      ],
      "measurement":[
        "total",
        "used"
      ]
    }
  }
}
```

Beachten Sie, dass Sie den Agenten bei jeder Änderung der Agentenkonfigurationsdatei neu starten müssen, damit die Änderungen wirksam werden.

Metriken, Protokolle und Ablaufverfolgungen an ein anderes Konto senden

Damit der CloudWatch Agent die Metriken, Logs oder Traces an ein anderes Konto sendet, geben Sie einen `role_arn` Parameter in der Agenten-Konfigurationsdatei auf dem sendenden Server an. Der `role_arn`-Wert legt eine IAM-Rolle im Zielkonto fest, die der Agent beim Senden von Daten an das Zielkonto verwendet. Diese Rolle ermöglicht dem sendenden Konto, eine entsprechende Rolle im Zielkonto anzunehmen, wenn die Metriken oder Protokolle für das Zielkonto bereitgestellt werden.

Sie können in der Konfigurationsdatei des Agenten auch separate `role_arn`-Zeichenfolgen angeben: eine für das Senden von Metriken, eine für das Senden von Protokollen und eine für das Senden von Ablaufverfolgungen.

Das folgende Beispiel für einen Teil des `agent`-Abschnitts in der Konfigurationsdatei legt fest, dass der Agent `CrossAccountAgentRole` verwendet, wenn er Daten an ein anderes Konto sendet.

```
{
  "agent": {
    "credentials": {
      "role_arn": "arn:aws:iam::123456789012:role/CrossAccountAgentRole"
    }
  },
  .....
```

```
}
```

Alternativ dazu können Sie im folgenden Beispiel verschiedene Rollen für das sendende Konto festlegen, um Metriken, Protokolle und Ablaufverfolgungen zu senden:

```
"metrics": {
    "credentials": {
     "role_arn": "RoleToSendMetrics"
    },
    "metrics_collected": {....
```

```
"logs": {
    "credentials": {
    "role_arn": "RoleToSendLogs"
    },
    ....
```

Erforderliche Richtlinien

Wenn Sie in der Agenten-Konfigurationsdatei einen Wert für `role_arn` festlegen, müssen Sie auch sicherstellen, dass die IAM-Rollen des sendenden Kontos und des Zielkontos über bestimmte Richtlinien verfügen. In den Rollen sowohl im sendenden als auch im Zielkonto muss `CloudWatchAgentServerPolicy` enthalten sein. Weitere Informationen zum Zuweisen dieser Richtlinie zu einer Rolle finden Sie unter Erstellen Sie IAM-Rollen zur Verwendung mit dem CloudWatch Agenten auf Amazon EC2 EC2-Instances.

Die Rolle im sendenden Konto muss außerdem die folgende Richtlinie enthalten. Sie fügen diese Richtlinie auf der Registerkarte Permissions (Berechtigungen) in der IAM-Konsole hinzu, wenn Sie die Rolle bearbeiten.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
```

```
            "Resource": [
                "arn:aws:iam::target-account-ID:role/agent-role-in-target-account"
            ]
        }
    ]
}
```

Die Rolle im Zielkonto muss die folgende Richtlinie enthalten, damit es die vom sendenden Konto verwendete IAM-Rolle erkennen kann. Sie fügen diese Richtlinie auf der Registerkarte Trust relationships (Vertrauensstellungen) in der IAM-Konsole hinzu, wenn Sie die Rolle bearbeiten. Die Rolle im Zielkonto, zu der Sie diese Richtlinie hinzufügen, ist die Rolle, die Sie unter Erstellen Sie IAM-Rollen und -Benutzer für die Verwendung mit dem Agenten CloudWatch erstellt haben. Dabei handelt es sich um die Rolle, die Sie unter *agent-role-in-target-account* in der Richtlinie angegebenen haben, die vom sendenden Konto verwendet wird.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::sending-account-ID:role/role-in-sender-account"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Zeitstempelunterschiede zwischen dem Unified CloudWatch Agent und dem früheren CloudWatch Logs-Agenten

Der CloudWatch Agent unterstützt im Vergleich zum früheren CloudWatch Logs-Agent einen anderen Satz von Symbolen für Zeitstempelformate. Diese Unterschiede sind in der folgenden Tabelle aufgeführt.

Von beiden Agenten unterstützte Symbole	Symbole, die nur von Unified CloudWatch Agent unterstützt werden	Symbole, die nur von früheren CloudWatch Logs-Agenten unterstützt wurden
%A, %a, %b, %B, %d, %f, %H, %I, %m, %M, %p, %S, %y, %Y, %Z, %z	%-d, %-I, %-m, %-M, %-S	%c,%j, %U, %W, %w

Weitere Informationen zu den Bedeutungen der vom neuen CloudWatch Agenten unterstützten Symbole finden Sie unter CloudWatch Agenten-Konfigurationsdatei: Abschnitt Protokolle im CloudWatch Amazon-Benutzerhandbuch. Informationen zu den vom CloudWatch Logs-Agenten unterstützten Symbolen finden Sie in der Agent-Konfigurationsdatei im Amazon CloudWatch Logs-Benutzerhandbuch.

Fehlerbehebung beim CloudWatch Agenten

Verwenden Sie die folgenden Informationen, um Probleme mit dem CloudWatch Agenten zu beheben.

Themen

- Den Agenten stoppen und neu starten CloudWatch

CloudWatch Befehlszeilenparameter für den Agenten

Um die vollständige Liste der vom CloudWatch Agenten unterstützten Parameter zu sehen, geben Sie in der Befehlszeile des Computers, auf dem der Agent installiert ist, Folgendes ein:

```
amazon-cloudwatch-agent-ctl -help
```

Die Installation des CloudWatch Agenten mithilfe von Run Command schlägt fehl

Um den CloudWatch Agenten mit Systems Manager Run Command zu installieren, muss der SSM-Agent auf dem Zielserver Version 2.2.93.0 oder höher sein. Wenn Ihr SSM Agent nicht die richtige Version aufweist, werden möglicherweise Fehler mit den folgenden Meldungen angezeigt:

```
no latest version found for package AmazonCloudWatchAgent on platform linux
```

```
failed to download installation package reliably
```

Informationen über das Installieren oder Aktualisieren der SSM-Agent-Version finden Sie unter Installieren und Konfigurieren des SSM-Agents in AWS Systems Manager -Benutzerhandbuch.

Der Agent lässt sich nicht starten CloudWatch

Wenn der CloudWatch Agent nicht gestartet werden kann, liegt möglicherweise ein Problem in Ihrer Konfiguration vor. Konfigurationsinformationen werden in die Datei `configuration-validation.log` geschrieben. Diese Datei befindet sich auf Linux-Servern unter `/opt/aws/amazon-cloudwatch-agent/logs/configuration-validation.log` und auf Servern mit Windows Server unter `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\configuration-validation.log`.

Stellen Sie sicher, dass der CloudWatch Agent läuft

Sie können den CloudWatch Agenten abfragen, um herauszufinden, ob er läuft oder angehalten wurde. Sie können AWS Systems Manager verwenden, um dies fernbedient zu erledigen. Sie können auch die Befehlszeile verwenden, aber damit nur den lokalen Server überprüfen.

Um den Status des CloudWatch Agenten mit Run Command abzufragen

1. Öffnen Sie die Systems Manager Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.

   –oder–

   Wenn die AWS Systems Manager Startseite geöffnet wird, scrollen Sie nach unten und wählen Sie Explore Run Command.
3. Wählen Sie Run Command (Befehl ausführen) aus.
4. Wählen Sie in der Liste der Befehlsdokumente die Schaltfläche neben AmazonCloudWatch-ausManageAgent.
5. Klicken Sie in der Liste Action auf Status.
6. Wählen Sie für Optional Configuration Source (Optionale Konfigurationsquelle) den Standardwert aus und lassen Sie Optional Configuration Location (Optionaler Konfigurationsstandort) leer.
7. Wählen Sie im Bereich Target die Instance, die Sie prüfen möchten.
8. Wählen Sie Ausführen aus.

Wenn der Agent ausgeführt wird, sieht die Ausgabe etwa folgendermaßen aus.

```
{
       "status": "running",
       "starttime": "2017-12-12T18:41:18",
       "version": "1.73.4"
}
```

Wenn der Agent angehalten ist, wird im Feld `"status"` `"stopped"` angezeigt.

Um den Status des CloudWatch Agenten lokal über die Befehlszeile abzufragen

- Geben Sie auf einem Linux-Server Folgendes ein:

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a
   status
  ```

  Geben Sie auf einem Server, auf dem Windows Server ausgeführt wird, PowerShell als Administrator Folgendes ein:

```
& $Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1 -m
 ec2 -a status
```

Der CloudWatch Agent startet nicht und der Fehler erwähnt eine Amazon EC2 EC2-Region

Wenn der Agent nicht startet und die Fehlermeldung einen Endpunkt einer Amazon-EC2-Region erwähnt, haben Sie den Agenten möglicherweise so konfiguriert, dass er Zugriff auf den Amazon-EC2-Endpunkt benötigt, ohne diesen Zugriff zu gewähren.

Beispiel: Wenn Sie einen Wert für den Parameter `append_dimensions` in der Agentenkonfigurationsdatei angeben, der von Amazon-EC2-Metadaten abhängt, und Sie Proxys verwenden, müssen Sie sicherstellen, dass der Server auf den Endpunkt für Amazon EC2 zugreifen kann. Weitere Informationen zu diesen Endpunkten finden Sie unter Amazon Elastic Compute Cloud (Amazon EC2) in der Allgemeine Amazon Web Services-Referenz.

Der CloudWatch Agent lässt sich auf Windows Server nicht starten

Unter Windows Server wird möglicherweise der folgende Fehler angezeigt:

```
Start-Service : Service 'Amazon CloudWatch Agent (AmazonCloudWatchAgent)' cannot be
 started due to the following
error: Cannot start service AmazonCloudWatchAgent on computer '.'.
At C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1:113
 char:12
+     $svc | Start-Service
+            ~~~~~~~~~~~~~
    + CategoryInfo          : OpenError:
 (System.ServiceProcess.ServiceController:ServiceController) [Start-Service],
   ServiceCommandException
    + FullyQualifiedErrorId :
 CouldNotStartService,Microsoft.PowerShell.Commands.StartServiceCommand
```

Um dies zu beheben, stellen Sie zunächst sicher, dass der Serverservice ausgeführt wird. Dieser Fehler kann angezeigt werden, wenn der Agent versucht, zu starten, während der Serverservice nicht ausgeführt wird.

Wenn der Serverservice bereits ausgeführt wird, kann Folgendes sein: Bei einigen Windows Server-Installationen benötigt der CloudWatch Agent mehr als 30 Sekunden, um zu starten. Da Windows Server standardmäßig nur 30 Sekunden zum Starten von Services zulässt, führt dies dazu, dass der Agent mit einem Fehler wie dem folgenden fehlschlägt:

Um dieses Problem zu beheben, erhöhen Sie den Zeitüberschreitungswert für Services. Weitere Informationen finden Sie unter Ein Service wird nicht gestartet und die Ereignisse 7000 und 7011 werden im Windows-Ereignisprotokoll protokolliert.

Wo sind die Metriken?

Wenn der CloudWatch Agent ausgeführt wurde, Sie aber keine von ihm gesammelten Messwerte im AWS Management Console oder im finden können, vergewissern Sie sich AWS CLI, dass Sie den richtigen Namespace verwenden. Der Namespace für die vom Agent zu erfassenden Metriken ist standardmäßig `CWAgent`. Sie können diesen Namespace mit dem Feld `namespace` im Abschnitt `metrics` der Agentenkonfigurationsdatei anpassen. Wenn Sie die erwarteten Metriken nicht finden, prüfen Sie die Konfigurationsdatei, um festzustellen, welchen Namespace Sie verwenden.

Wenn Sie das CloudWatch Agentenpaket zum ersten Mal herunterladen, befindet `amazon-cloudwatch-agent.json` sich die Agenten-Konfigurationsdatei. Diese Datei befindet sich in dem Verzeichnis, in dem Sie den Konfigurationsassistenten ausgeführt haben. Möglicherweise haben Sie die Datei in ein anderes Verzeichnis verschoben. Wenn Sie den Konfigurations-Assistenten verwenden, wird die Ausgabe der Agent-Konfigurationsdatei vom Assistenten mit dem Namen `config.json` versehen. Weitere Informationen zur Konfigurationsdatei, einschließlich des Felds `namespace`, finden Sie unter CloudWatch Agenten-Konfigurationsdatei: Abschnitt „Metriken“.

Es dauert lange, bis der CloudWatch Agent in einem Container ausgeführt wird, oder es wird ein Hop-Limit-Fehler protokolliert

Wenn Sie den CloudWatch Agenten als Container-Service ausführen und Amazon EC2-Metrikdimensionen zu allen vom Agenten gesammelten Metriken hinzufügen möchten, werden in Version v1.247354.0 des Agenten möglicherweise die folgenden Fehler angezeigt:

```
2022-06-07T03:36:11Z E! [processors.ec2tagger] ec2tagger: Unable to retrieve Instance
 Metadata Tags. This plugin must only be used on an EC2 instance.
2022-06-07T03:36:11Z E! [processors.ec2tagger] ec2tagger: Please increase hop limit
 to 2 by following this document https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/
configuring-instance-metadata-options.html#configuring-IMDS-existing-instances.
```

```
2022-06-07T03:36:11Z E! [telegraf] Error running agent: could not initialize processor
 ec2tagger: EC2MetadataRequestError: failed to get EC2 instance identity document
caused by: EC2MetadataError: failed to make EC2Metadata request
        status code: 401, request id:
caused by:
```

Dieser Fehler kann auftreten, wenn der Agent versucht, Metadaten von IMDSv2 innerhalb eines Containers ohne entsprechendes Hop-Limit abzurufen. In Agentenversionen vor v1.247354.0 kann dieses Problem auftreten, ohne dass die Protokollmeldung angezeigt wird.

Erhöhen Sie zur Lösung dieses Problems das Hop-Limit auf „2“. Eine entsprechende Anleitung finden Sie unter Konfigurieren der Optionen für Instance-Metadaten.

Ich habe meine Agentenkonfiguration aktualisiert, sehe aber die neuen Metriken oder Protokolle nicht in der Konsole CloudWatch

Wenn Sie Ihre CloudWatch Agenten-Konfigurationsdatei aktualisieren, müssen Sie beim nächsten Start des Agenten die **`fetch-config`** Option verwenden. Wenn Sie beispielsweise die aktualisierte Datei auf dem lokalen Computer gespeichert haben, geben Sie den folgenden Befehl ein:

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -
s -m ec2 -c file:configuration-file-path
```

CloudWatch Agentendateien und Speicherorte

In der folgenden Tabelle sind die vom CloudWatch Agenten installierten und mit ihm verwendeten Dateien sowie ihre Speicherorte auf Servern aufgeführt, auf denen Linux oder Windows Server ausgeführt werden.

Datei	Linux-Speicherort	Windows Server-Speicherort
Das Steuerskript, das das Starten, Anhalten und Neustarten des Agents kontrolliert.	`/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl` oder `/usr/bin/amazon-cloudwatch-agent-ctl`	`$Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1`

Datei	Linux-Speicherort	Windows Server-Speicherort
Die Protokolldatei, in die der Agent schreibt. Möglicher weise müssen Sie dies bei der Kontaktaufnahme anhängen AWS Support.	`/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log` oder `/var/log/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.log`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\amazon-cloudwatch-agent.log`
Datei für die Auswertung der Agentenkonfiguration.	`/opt/aws/amazon-cloudwatch-agent/logs/configuration-validation.log` oder `/var/log/amazon/amazon-cloudwatch-agent/configuration-validation.log`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\configuration-validation.log`
Die für die Konfiguration des Agenten verwendete JSON-Datei, sofort nach der Erstellung durch den Assistenten. Weitere Informationen finden Sie unter Erstellen Sie die CloudWatch Agent-Konfigurationsdatei.	`/opt/aws/amazon-cloudwatch-agent/bin/config.json`	`$Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\config.json`
Die für die Konfiguration des Agenten verwendete JSON-Datei, wenn diese Konfigurationsdatei aus Parameter Store heruntergeladen wurde.	`/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json` oder `/etc/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json`

Datei	Linux-Speicherort	Windows Server-Speicherort
Die TOML-Datei, mit der Regions- und Anmeldeinformationen angegeben werden, die vom Agenten verwendet werden sollen, wobei die Systemvorgaben überschrieben werden.	`/opt/aws/amazon-cloudwatch-agent/etc/common-config.toml` oder `/etc/amazon/amazon-cloudwatch-agent/common-config.toml`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\common-config.toml`
Die TOML-Datei, die den konvertierten Inhalt der JSON-Konfigurationsdatei enthält. Das `amazon-cloudwatch-agent-ctl` -Skript generiert diese Datei. Benutzer sollten diese Datei nicht direkt ändern. Sie kann nützlich sein, um zu überprüfen, ob die Übersetzung von JSON in TOML erfolgreich war.	`/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml` oder `/etc/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.toml`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.toml`
Die YAML-Datei, die den konvertierten Inhalt der JSON-Konfigurationsdatei enthält. Das `amazon-cloudwatch-agent-ctl` -Skript generiert diese Datei. Sie sollten diese Datei nicht direkt ändern. Diese Datei kann nützlich sein, um zu überprüfen, ob die Übersetzung von JSON nach YAML erfolgreich war.	`/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.yaml or /etc/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.yaml`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.yaml`

Suchen Sie nach Informationen zu CloudWatch Agentenversionen

Geben Sie den folgenden Befehl ein, um die Versionsnummer des CloudWatch Agenten auf einem Linux-Server zu ermitteln:

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a status
```

Geben Sie den folgenden Befehl ein, um die Versionsnummer des CloudWatch Agenten auf Windows Server zu ermitteln:

```
& $Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1 -m ec2
 -a status
```

> **Note**
>
> Die Verwendung dieses Befehls ist der richtige Weg, um die Version des CloudWatch Agenten zu finden. Wenn Sie Programme und Feature nutzen, sehen Sie in der Systemsteuerung eine falsche Versionsnummer.

Sie können auch eine README-Datei über die neuesten Änderungen am Agenten sowie eine Datei mit der Versionsnummer herunterladen, die aktuell zum Download verfügbar ist. Diese Dateien befinden sich an den folgenden Speicherorten:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/RELEASE_NOTES` oder `https://amazoncloudwatch-agent-`*`region`*`.s3.`*`region`*`.amazonaws.com/info/latest/RELEASE_NOTES`
- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/CWAGENT_VERSION` oder `https://amazoncloudwatch-agent-`*`region`*`.s3.`*`region`*`.amazonaws.com/info/latest/CWAGENT_VERSION`

Vom CloudWatch Agenten generierte Protokolle

Der Agent generiert ein Protokoll, während es ausgeführt wird. Dieses Protokoll enthält Informationen zur Fehlerbehebung. Das Protokoll ist die Datei `amazon-cloudwatch-agent.log`. Diese Datei befindet sich auf Linux-Servern unter `/opt/aws/amazon-cloudwatch-agent/logs/amazon-`

`cloudwatch-agent.log` und auf Servern mit Windows Server unter `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\amazon-cloudwatch-agent.log`.

Sie können den Agenten so konfigurieren, dass zusätzliche Details in der Datei `amazon-cloudwatch-agent.log` protokolliert werden. Setzen Sie in der Agentenkonfigurationsdatei im `agent` Abschnitt das `debug` Feld auf`true`, konfigurieren Sie den CloudWatch Agenten neu und starten Sie ihn neu. Um die Aufzeichnung dieser zusätzlichen Informationen zu deaktivieren, setzen Sie das Feld `debug` auf `false`. Konfigurieren Sie dann den Agenten neu und starten Sie ihn neu. Weitere Informationen finden Sie unter Erstellen oder bearbeiten Sie die CloudWatch Agenten-Konfigurationsdatei manuell.

In den Versionen 1.247350.0 und höher des CloudWatch Agenten können Sie das `aws_sdk_log_level` Feld im `agent` Abschnitt der Agenten-Konfigurationsdatei optional auf eine oder mehrere der folgenden Optionen festlegen. Trennen Sie mehrere Optionen mit dem `|`-Zeichen.

- `LogDebug`
- `LogDebugWithSigning`
- `LogDebugWithHTTPBody`
- `LogDebugRequestRetries`
- `LogDebugWithEventStreamBody`

Weitere Informationen zu diesen Optionen finden Sie unter. LogLevelType

Den Agenten stoppen und neu starten CloudWatch

Sie können den CloudWatch Agenten manuell entweder über die Befehlszeile AWS Systems Manager oder über die Befehlszeile beenden.

Um den CloudWatch Agenten mit Run Command zu beenden

1. Öffnen Sie die Systems Manager Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.

   –oder–

   Wenn die AWS Systems Manager Startseite geöffnet wird, scrollen Sie nach unten und wählen Sie Explore Run Command.

3. Wählen Sie Run Command (Befehl ausführen) aus.
4. Wählen Sie in der Command-Dokumentliste die Option AmazonCloudWatch- ManageAgent.
5. Wählen Sie im Bereich Ziele die Instanz aus, auf der Sie den CloudWatch Agenten installiert haben.
6. Klicken Sie in der Liste Action auf stop.
7. Lassen Sie Optional Configuration Source (Optionale Konfigurationsquelle) und Optional Configuration Location (Optionaler Konfigurationsstandort) leer.
8. Wählen Sie Ausführen aus.

Um den CloudWatch Agenten lokal über die Befehlszeile zu beenden

- Geben Sie auf einem Linux-Server Folgendes ein:

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a
   stop
  ```

  Geben Sie auf einem Server, auf dem Windows Server ausgeführt wird, PowerShell als Administrator Folgendes ein:

  ```
  & $Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1 -m
   ec2 -a stop
  ```

Um den Agent neu zu starten, befolgen Sie die Anweisungen in Starten Sie den Agenten CloudWatch .

Einbetten von Metriken in Protokollen

Das CloudWatch eingebettete Metrikformat ermöglicht es Ihnen, benutzerdefinierte Metriken asynchron in Form von Protokollen zu generieren, die in Logs geschrieben werden. CloudWatch Sie können benutzerdefinierte Metriken zusammen mit detaillierten Protokolldaten einbetten und die benutzerdefinierten Metriken CloudWatch automatisch extrahieren, sodass Sie sie visualisieren und als Alarm auslösen können, um Vorfälle in Echtzeit zu erkennen. Darüber hinaus können die detaillierten Protokollereignisse, die mit den extrahierten Metriken verknüpft sind, mithilfe von CloudWatch Logs Insights abgefragt werden, um tiefe Einblicke in die Hauptursachen von Betriebsereignissen zu erhalten.

Das eingebettete Metrikformat hilft Ihnen, verwertbare benutzerdefinierte Metriken aus flüchtigen Ressourcen wie Lambda-Funktionen und -Containern zu generieren. Durch das eingebettete Metrikformat zum Senden von Protokollen aus diesen flüchtigen Ressourcen können Sie jetzt ganz einfach benutzerdefinierte Metriken erstellen, ohne separaten Code instrumentieren oder verwalten zu müssen, während Sie leistungsstarke Analysefunktionen für Ihre Protokolldaten erhalten.

Für die Verwendung des eingebetteten metrischen Formats ist keine Einrichtung erforderlich. Strukturieren Sie Ihre Logs entweder gemäß der Formatspezifikation für eingebettete Metriken oder generieren Sie sie mithilfe unserer Client-Bibliotheken und senden Sie sie über die PutLogEvents API oder den CloudWatch Agenten an CloudWatch Logs.

Es fallen Gebühren für die Protokollaufnahme und -Archivierung sowie benutzerdefinierte Metriken an, die generiert werden. Weitere Informationen finden Sie unter Amazon CloudWatch – Preise.

> **Note**
>
> Seien Sie vorsichtig, wenn Sie die Metrikextraktion konfigurieren, da sich dies auf Ihre benutzerdefinierte Metrikauslastung und die entsprechende Rechnung auswirkt. Wenn Sie versehentlich Metriken basierend auf hohen Kardinalitätsdimensionen erstellen (z. B. `requestId`), erstellt das eingebettete Metrikformat standardmäßig eine benutzerdefinierte Metrik, die jeder eindeutigen Dimensionskombination entspricht. Weitere Informationen finden Sie unter Dimensionen.

Themen

- Anzeigen Ihrer Metriken und Protokolle in der Konsole
- Alarme für Metriken setzen, die mit dem eingebetteten Metrikformat erstellt wurden

Veröffentlichen von Protokollen mithilfe des eingebetteten Metrikformats

Sie können Protokolle im eingebetteten Metrikformat mit den folgenden Methoden generieren:

- Generieren und Senden der Protokolle mithilfe der Open-Source-Client-Bibliotheken.
- Generieren Sie die Protokolle manuell mithilfe der Spezifikation für das eingebettete metrische Format und verwenden Sie dann den CloudWatch Agenten oder die PutLogEvents API, um die Protokolle zu senden.

Themen

- Erstellen von Protokollen im eingebetteten Metrikformat unter Verwendung der Client-Bibliotheken
- Spezifikation: Eingebettetes Metrikformat
- Verwenden der PutLogEvents API zum Senden manuell erstellter Logs im eingebetteten metrischen Format
- Verwenden des CloudWatch Agenten zum Senden eingebetteter Logs im Metrikformat
- Verwenden des eingebetteten metrischen Formats mit AWS Distro für OpenTelemetry

Erstellen von Protokollen im eingebetteten Metrikformat unter Verwendung der Client-Bibliotheken

Amazon stellt Open-Source-Client-Bibliotheken bereit, mit denen Sie Protokolle im eingebetteten Metrikformat erstellen können. Derzeit sind diese Bibliotheken für die Sprachen in der folgenden Liste verfügbar. Vollständige Beispiele für verschiedene Einrichtungen finden Sie in unseren Client-Bibliotheken unter /examples.

Die Bibliotheken und Anweisungen zur Verwendung befinden sich auf Github. Verwenden Sie die folgenden Links.

- Node.Js

> **Note**
>
> Für Node.js sind die Versionen 4.1.1+, 3.0.2+, 2.0.7+ für die Verwendung mit dem Lambda-JSON-Protokollformat erforderlich. Die Verwendung früherer Versionen in solchen Lambda-Umgebungen führt zu Metrik-Verlusten.
> Weitere Informationen finden Sie unter Zugreifen auf CloudWatch Amazon-Protokolle für AWS Lambda.

- Python
- Java
- C#

Client-Bibliotheken sind so konzipiert, dass sie sofort mit dem CloudWatch Agenten zusammenarbeiten. Generierte Logs im eingebetteten metrischen Format werden an den CloudWatch Agenten gesendet, der sie dann aggregiert und für Sie in CloudWatch Logs veröffentlicht.

> **Note**
>
> Bei der Verwendung von Lambda ist kein Agent erforderlich, an den die Protokolle gesendet CloudWatch werden. Alles, was in STDOUT protokolliert wurde, wird über den Lambda CloudWatch Logging Agent an Logs gesendet.

Spezifikation: Eingebettetes Metrikformat

Das CloudWatch eingebettete metrische Format ist eine JSON-Spezifikation, die verwendet wird, um CloudWatch Logs anzuweisen, Metrikwerte, die in strukturierte Protokollereignisse eingebettet sind, automatisch zu extrahieren. Sie können CloudWatch es verwenden, um die extrahierten Metrikwerte grafisch darzustellen und Alarme zu erstellen.

Konventionen der Spezifikationen für eingebettete Metrikformate

Die Schlüsselwörter „MUSS“, „DARF NICHT“, „NOTWENDIG“, „SOLL“, „SOLL NICHT“, „SOLLTE“, „SOLLTE NICHT“, „EMPFOHLEN“, „KANN“ und „OPTIONAL“ in dieser Formatspezifikation sind wie in den Key Words RFC2119 beschrieben zu interpretieren.

Die Begriffe „JSON“, „JSON-Text“, „JSON-Wert“, „Mitglied“, „Element“, „Objekt“, „Array“, „Zahl“, „Zeichenfolge“, „Boolean“, „wahr“, „falsch“ und „Null“ in dieser Formatspezifikation sind so zu interpretieren, wie sie in JavaScript Object Notation RFC8259 definiert sind.

> **Note**
>
> Wenn Sie Alarme für Metriken erstellen möchten, die im eingebetteten Metrikformat erstellt wurden, siehe Alarme für Metriken setzen, die mit dem eingebetteten Metrikformat erstellt wurden für Empfehlungen.

Eingebettetes Metrikformat-Dokumentstruktur

Dieser Abschnitt beschreibt die Struktur eines Dokuments im eingebetteten Metrikformat. Dokumente im eingebetteten metrischen Format sind in Object Notation RFC8259 definiert. JavaScript

Sofern nicht anders angegeben, DÜRFEN in dieser Spezifikation definierte Objekte KEINE zusätzlichen Elemente enthalten. Elemente, die von dieser Spezifikation nicht erkannt werden, MÜSSEN ignoriert werden. In dieser Spezifikation definierte Elemente berücksichtigen Groß- und Kleinschreibung.

Das eingebettete metrische Format unterliegt denselben Beschränkungen wie standardmäßige CloudWatch Log-Ereignisse und ist auf eine maximale Größe von 256 KB begrenzt.

Mit dem eingebetteten Metrikformat können Sie die Verarbeitung Ihrer EMF-Protokolle nach Metriken verfolgen, die im `AWS/Logs`-Namespace Ihres Kontos veröffentlicht werden. Diese können verwendet werden, um die fehlgeschlagene Metrikgenerierung von EMF nachzuverfolgen und festzustellen, ob Fehler aufgrund der Analyse oder der Validierung auftreten. Weitere Informationen finden Sie unter Überwachung mit CloudWatch Metriken.

Stammknoten

Die LogEvent Nachricht MUSS ein gültiges JSON-Objekt ohne zusätzliche Daten am Anfang oder Ende der LogEvent Nachrichtenzeichenfolge sein. Weitere Hinweise zur LogEvent Struktur finden Sie unter InputLogEvent.

Dokumente im eingebetteten Metrikformat MÜSSEN das folgende Element der obersten Ebene auf dem Stammknoten enthalten. Dies ist ein Metadatenobjekt-Objekt.

```
{
 "_aws": {
```

```
    "CloudWatchMetrics": [ ... ]
  }
}
```

Der Stammknoten MUSS alle Zielmitglieder-Elemente enthalten, die durch die Verweise in MetricDirective Objekt definiert sind.

Der Stammknoten KANN alle anderen Elemente enthalten, die nicht in den oben genannten Anforderungen enthalten sind. Die Werte dieser Elemente MÜSSEN gültige JSON-Typen sein.

Metadatenobjekt

Das `_aws` Element kann verwendet werden, um Metadaten über die Nutzlast darzustellen, die nachgelagerten Dienste darüber informieren, wie sie diese verarbeiten sollen. LogEvent Der Wert MUSS ein Objekt sein und MUSS die folgenden Elemente enthalten:

- CloudWatchMetrics— Ein Array von, das MetricDirective Objekt verwendet wird, um anzuweisen CloudWatch , Metriken aus dem Stammknoten des zu extrahieren. LogEvent

```
{
  "_aws": {
    "CloudWatchMetrics": [ ... ]
  }
}
```

- Zeitstempel – Eine Zahl, die den Zeitstempel für Metriken darstellt, die aus dem Ereignis extrahiert werden. Werte MÜSSEN als die Anzahl der Millisekunden nach dem 1. Januar 1970 00:00:00 UTC ausgedrückt werden.

```
{
  "_aws": {
    "Timestamp": 1559748430481
  }
}
```

MetricDirective Objekt

Das MetricDirective Objekt weist nachgelagerte Dienste an, dass es Metriken LogEvent enthält, die extrahiert und veröffentlicht werden. CloudWatch MetricDirectives MUSS die folgenden Mitglieder enthalten:

- Namespace — Eine Zeichenfolge, die den CloudWatch Namespace für die Metrik darstellt.
- Dimensionen – Ein DimensionSet Array.
- Metriken– – Ein Array von MetricDefinition-Objekten. Dieses Array DARF NICHT mehr als 100 MetricDefinition Objekte enthalten.

DimensionSet Array

A DimensionSet ist ein Array von Zeichenketten, die die Dimensionsschlüssel enthalten, die auf alle Metriken im Dokument angewendet werden. Die Werte innerhalb dieses Arrays MÜSSEN auch Elemente auf dem Stammknoten sein, die als das Zielmitglieder bezeichnet werden.

A DimensionSet DARF NICHT mehr als 30 Dimensionsschlüssel enthalten. A DimensionSet KANN leer sein.

Das Zielelement MUSS einen Zeichenfolgenwert haben. Dieser Wert DARF NICHT mehr als 1024 Zeichen enthalten. Das Zielelement definiert eine Dimension, die als Teil der Metrikidentität veröffentlicht wird. Jeder DimensionSet Benutzer erstellt eine neue Metrik in CloudWatch. Weitere Informationen zu Dimensionen finden Sie unter Dimension und Dimensionen.

```
{
 "_aws": {
   "CloudWatchMetrics": [
     {
       "Dimensions": [ [ "functionVersion" ] ],
       ...
     }
   ]
 },
 "functionVersion": "$LATEST"
}
```

> **Note**
>
> Seien Sie vorsichtig, wenn Sie die Metrikextraktion konfigurieren, da sich dies auf Ihre benutzerdefinierte Metrikauslastung und die entsprechende Rechnung auswirkt. Wenn Sie versehentlich Metriken basierend auf hohen Kardinalitätsdimensionen erstellen (z. B. `requestId`), erstellt das eingebettete Metrikformat standardmäßig eine benutzerdefinierte

Metrik, die jeder eindeutigen Dimensionskombination entspricht. Weitere Informationen finden Sie unter Dimensionen.

MetricDefinition Objekt

A MetricDefinition ist ein Objekt, das das folgende Mitglied enthalten MUSS:

- Name – Eine Zeichenfolge Referenzwerte auf eine Metrik Zielmitglieder. Metrikziele MÜSSEN entweder aus einem numerischen Wert oder einem Array von numerischen Werten bestehen.

Ein MetricDefinition Objekt KANN die folgenden Mitglieder enthalten:

- Einheit – Ein OPTIONALER Zeichenfolgenwert, der die Maßeinheit für die entsprechende Metrik darstellt. Die Werte SOLLTEN gültige CloudWatch metrische Einheiten sein. Hinweise zu gültigen Einheiten finden Sie unter MetricDatum. Wenn kein Wert angegeben wird, wird der Standardwert NONE angenommen.
- StorageResolution— Ein OPTIONALER Ganzzahlwert, der die Speicherauflösung für die entsprechende Metrik darstellt. Wenn dieser Wert auf 1 gesetzt wird, handelt es sich bei dieser Metrik um eine Metrik mit hoher Auflösung, sodass die Metrik mit einer Auflösung von unter einer Minute bis zu einer Sekunde CloudWatch gespeichert wird. Wenn dieser Wert auf 60 gesetzt wird, handelt es sich bei dieser Metrik um eine Standardauflösung, die mit einer Auflösung von 1 Minute CloudWatch gespeichert wird. Die Werte SOLLTEN gültige CloudWatch unterstützte Auflösungen sein, 1 oder 60. Wenn kein Wert angegeben wird, wird der Standardwert 60 angenommen.

  Weitere Informationen zu hochauflösenden Metriken finden Sie unter Hochauflösende Metriken.

> **Note**
>
> Wenn Sie Alarme für Metriken erstellen möchten, die im eingebetteten Metrikformat erstellt wurden, siehe Alarme für Metriken setzen, die mit dem eingebetteten Metrikformat erstellt wurden für Empfehlungen.

```
{
  "_aws": {
    "CloudWatchMetrics": [
```

```
      {
        "Metrics": [
          {
            "Name": "Time",
            "Unit": "Milliseconds",
            "StorageResolution": 60
          }
        ],
        ...
      }
    ]
  },
  "Time": 1
}
```

Referenzwerte

Referenzwerte sind Zeichenfolgenwerte, die Zielmitglieder-Elemente auf dem Stammknoten referenzieren. Diese Referenzen sollten NICHT mit den in RFC6901 beschriebenen JSON Pointers verwechselt werden. Zielwerte können nicht verschachtelt werden.

Zielmitglieder

Gültige Ziele MÜSSEN Elemente auf dem Stammknoten sein und dürfen keine verschachtelten Objekte sein. Ein _reference_-Wert von `"A.a"` MUSS beispielsweise mit dem folgenden Element übereinstimmen:

```
{ "A.a" }
```

Er DARF NICHT mit dem verschachtelten Element übereinstimmen:

```
{ "A": { "a" } }
```

Gültige Werte von Zielelementen hängen davon ab, was sie referenziert. Ein Metrikziel MUSS ein numerischer Wert oder ein Array numerischer Werte sein. Numerische Array-Metrikziele DÜRFEN NICHT mehr als 100 Mitglieder haben. Ein Dimensionsziel MUSS einen Zeichenfolgenwert haben.

Beispiel für ein eingebettetes Metrikformat und JSON-Schema

Im Folgenden finden Sie ein gültiges Beispiel für ein eingebettetes Metrikformat.

```
{
```

```
  "_aws": {
    "Timestamp": 1574109732004,
    "CloudWatchMetrics": [
      {
        "Namespace": "lambda-function-metrics",
        "Dimensions": [["functionVersion"]],
        "Metrics": [
          {
            "Name": "time",
            "Unit": "Milliseconds",
            "StorageResolution": 60
          }
        ]
      }
    ]
  },
  "functionVersion": "$LATEST",
  "time": 100,
  "requestId": "989ffbf8-9ace-4817-a57c-e4dd734019ee"
}
```

Sie können das folgende Schema verwenden, um Dokumente im eingebetteten Metrikformat zu validieren.

```
{
    "type": "object",
    "title": "Root Node",
    "required": [
        "_aws"
    ],
    "properties": {
        "_aws": {
            "$id": "#/properties/_aws",
            "type": "object",
            "title": "Metadata",
            "required": [
                "Timestamp",
                "CloudWatchMetrics"
            ],
            "properties": {
                "Timestamp": {
                    "$id": "#/properties/_aws/properties/Timestamp",
                    "type": "integer",
```

```
                    "title": "The Timestamp Schema",
                    "examples": [
                        1565375354953
                    ]
                },
                "CloudWatchMetrics": {
                    "$id": "#/properties/_aws/properties/CloudWatchMetrics",
                    "type": "array",
                    "title": "MetricDirectives",
                    "items": {
                        "$id": "#/properties/_aws/properties/CloudWatchMetrics/items",
                        "type": "object",
                        "title": "MetricDirective",
                        "required": [
                            "Namespace",
                            "Dimensions",
                            "Metrics"
                        ],
                        "properties": {
                            "Namespace": {
                                "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Namespace",
                                "type": "string",
                                "title": "CloudWatch Metrics Namespace",
                                "examples": [
                                    "MyApp"
                                ],
                                "pattern": "^(.*)$",
                                "minLength": 1,
                                "maxLength": 1024
                            },
                            "Dimensions": {
                                "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Dimensions",
                                "type": "array",
                                "title": "The Dimensions Schema",
                                "minItems": 1,
                                "items": {
                                    "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Dimensions/items",
                                    "type": "array",
                                    "title": "DimensionSet",
                                    "minItems": 0,
                                    "maxItems": 30,
```

```
                                    "items": {
                                        "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Dimensions/items/items",
                                        "type": "string",
                                        "title": "DimensionReference",
                                        "examples": [
                                            "Operation"
                                        ],
                                        "pattern": "^(.*)$",
                                        "minLength": 1,
                                        "maxLength": 250
}
                                }
                            },
                            "Metrics": {
                                "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Metrics",
                                "type": "array",
                                "title": "MetricDefinitions",
                                "items": {
                                    "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items",
                                    "type": "object",
                                    "title": "MetricDefinition",
                                    "required": [
                                        "Name"
                                    ],
                                    "properties": {
                                        "Name": {
                                            "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items/properties/Name",
                                            "type": "string",
                                            "title": "MetricName",
                                            "examples": [
                                                "ProcessingLatency"
                                            ],
                                            "pattern": "^(.*)$",
                                            "minLength": 1,
                                            "maxLength": 1024
                                        },
                                        "Unit": {
                                            "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items/properties/Unit",
                                            "type": "string",
```

```
                                            "title": "MetricUnit",
                                            "examples": [
                                                "Milliseconds"
                                            ],
                                            "pattern": "^(Seconds|Microseconds|
Milliseconds|Bytes|Kilobytes|Megabytes|Gigabytes|Terabytes|Bits|Kilobits|Megabits|
Gigabits|Terabits|Percent|Count|Bytes\\/Second|Kilobytes\\/Second|Megabytes\\/Second|
Gigabytes\\/Second|Terabytes\\/Second|Bits\\/Second|Kilobits\\/Second|Megabits\\/
Second|Gigabits\\/Second|Terabits\\/Second|Count\\/Second|None)$"
                                        },
                                        "StorageResolution": {
                                            "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items/properties/StorageResolution",
                                            "type": "integer",
                                            "title": "StorageResolution",
                                            "examples": [
                                                60
                                            ]
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

Verwenden der PutLogEvents API zum Senden manuell erstellter Logs im eingebetteten metrischen Format

Mithilfe der PutLogEvents Logs-API können Sie eingebettete Logs im metrischen Format an CloudWatch CloudWatch Logs senden. Beim Aufrufen können Sie optional den folgenden HTTP-Header angeben PutLogEvents, um CloudWatch Logs anzuweisen, dass die Metriken extrahiert werden sollen. Dies ist jedoch nicht mehr erforderlich.

```
x-amzn-logs-format: json/emf
```

Im Folgenden finden Sie ein vollständiges Beispiel für die Verwendung des AWS SDK for Java 2.x:

```
package org.example.basicapp;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import software.amazon.awssdk.services.cloudwatchlogs.model.DescribeLogStreamsRequest;
import software.amazon.awssdk.services.cloudwatchlogs.model.DescribeLogStreamsResponse;
import software.amazon.awssdk.services.cloudwatchlogs.model.InputLogEvent;
import software.amazon.awssdk.services.cloudwatchlogs.model.PutLogEventsRequest;

import java.util.Collections;

public class EmbeddedMetricsExample {
        public static void main(String[] args) {

                final String usage = "To run this example, supply a Region code (eg.
 us-east-1), log group, and stream name as command line arguments"
                                + "Ex: PutLogEvents <region-id> <log-group-name>
 <stream-name>";

                if (args.length != 3) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String regionId = args[0];
                String logGroupName = args[1];
                String logStreamName = args[2];

                CloudWatchLogsClient logsClient =
 CloudWatchLogsClient.builder().region(Region.of(regionId)).build();

                // Build a JSON log using the EmbeddedMetricFormat.
                long timestamp = System.currentTimeMillis();
                String message = "{" +
                                "  \"_aws\": {" +
                                "    \"Timestamp\": " + timestamp  + "," +
                                "    \"CloudWatchMetrics\": [" +
                                "      {" +
                                "        \"Namespace\": \"MyApp\"," +
                                "        \"Dimensions\": [[\"Operation\"], [\"Operation
\", \"Cell\"]]," +
                                "        \"Metrics\": [{ \"Name\": \"ProcessingLatency
\", \"Unit\": \"Milliseconds\", \"StorageResolution\": 60 }]" +
```

```
                                "      }" +
                                "    ]" +
                                "  }," +
                                "  \"Operation\": \"Aggregator\"," +
                                "  \"Cell\": \"001\"," +
                                "  \"ProcessingLatency\": 100" +
                                "}";
                InputLogEvent inputLogEvent = InputLogEvent.builder()
                        .message(message)
                        .timestamp(timestamp)
                        .build();

                // Specify the request parameters.
                PutLogEventsRequest putLogEventsRequest = PutLogEventsRequest.builder()
                        .logEvents(Collections.singletonList(inputLogEvent))
                        .logGroupName(logGroupName)
                        .logStreamName(logStreamName)
                        .build();

                logsClient.putLogEvents(putLogEventsRequest);

                System.out.println("Successfully put CloudWatch log event");
        }

}
```

> **Note**
>
> Mit dem eingebetteten Metrikformat können Sie die Verarbeitung Ihrer EMF-Protokolle nach Metriken verfolgen, die im `AWS/Logs`-Namespace Ihres Kontos veröffentlicht werden. Diese können verwendet werden, um die fehlgeschlagene Metrikgenerierung von EMF nachzuverfolgen und festzustellen, ob Fehler aufgrund der Analyse oder der Validierung auftreten. Weitere Informationen finden Sie unter Überwachung mit CloudWatch Metriken.

Verwenden des CloudWatch Agenten zum Senden eingebetteter Logs im Metrikformat

Um diese Methode zu verwenden, installieren Sie zuerst den CloudWatch Agenten für die Dienste, von denen Sie Logs im eingebetteten metrischen Format senden möchten, und dann können Sie mit dem Senden der Ereignisse beginnen.

Der CloudWatch Agent muss Version 1.230621.0 oder höher sein.

> **Note**
>
> Sie müssen den CloudWatch Agenten nicht installieren, um Protokolle von Lambda-Funktionen zu senden.
> Lambda-Funktionstimeouts werden nicht automatisch behandelt. Dies bedeutet, dass die Metriken für diesen Aufruf nicht erfasst werden, wenn für Ihre Funktion eine Zeitüberschreitung auftritt, bevor die Metriken übertragen werden.

Den Agenten installieren CloudWatch

Installieren Sie den CloudWatch Agenten für jeden Dienst, der eingebettete Protokolle im metrischen Format senden soll.

Den CloudWatch Agenten auf EC2 installieren

Installieren Sie zunächst den CloudWatch Agenten auf der Instance. Weitere Informationen finden Sie unter Den CloudWatch Agenten installieren.

Nachdem Sie den Agenten installiert haben, konfigurieren Sie den Agenten so, dass er einen UDP- oder TCP-Port auf die Protokolle im eingebetteten Metrikformat überwacht. Im Folgenden finden Sie ein Beispiel für diese Konfiguration, die den Standard-Socket `tcp:25888` überwacht. Weitere Informationen zur Agentenkonfiguration finden Sie unter Erstellen oder bearbeiten Sie die CloudWatch Agenten-Konfigurationsdatei manuell

```
{
  "logs": {
    "metrics_collected": {
      "emf": { }
    }
  }
}
```

Installation des CloudWatch Agenten auf Amazon ECS

Der einfachste Weg, den CloudWatch Agenten auf Amazon ECS bereitzustellen, besteht darin, ihn als Sidecar auszuführen und ihn in derselben Aufgabendefinition wie Ihre Anwendung zu definieren.

Erstellen der Agentenkonfigurationsdatei

Erstellen Sie Ihre CloudWatch Agenten-Konfigurationsdatei lokal. In diesem Beispiel lautet der relative Dateipfad `amazon-cloudwatch-agent.json`.

Weitere Informationen zur Agentenkonfiguration finden Sie unter Erstellen oder bearbeiten Sie die CloudWatch Agenten-Konfigurationsdatei manuell

```
{
  "logs": {
    "metrics_collected": {
      "emf": { }
    }
  }
}
```

Übertragen der Konfiguration an den SSM-Parameterspeicher

Geben Sie den folgenden Befehl ein, um die CloudWatch Agentenkonfigurationsdatei in den AWS Systems Manager (SSM) -Parameterspeicher zu übertragen.

```
aws ssm put-parameter \
    --name "cwagentconfig" \
    --type "String" \
    --value "`cat amazon-cloudwatch-agent.json`" \
    --region "{{region}}"
```

Konfigurieren der Aufgabendefinition

Konfigurieren Sie Ihre Aufgabendefinition so, dass sie den CloudWatch Agenten verwendet und den TCP- oder UDP-Port verfügbar macht. Die Beispielaufgabendefinition, die Sie verwenden sollten, hängt vom Netzwerkmodus ab.

Beachten Sie, dass die `webapp` die `AWS_EMF_AGENT_ENDPOINT`-Umgebungsvariable angibt. Diese wird von der Bibliothek verwendet und sollte auf den Endpunkt verweisen, den der Agent überwacht. Darüber hinaus gibt der `cwagent` den `CW_CONFIG_CONTENT` als „valueFrom“-Parameter an, der auf die SSM-Konfiguration verweist, die Sie im vorherigen Schritt erstellt haben.

Dieser Abschnitt enthält ein Beispiel für den Bridge-Modus und ein Beispiel für den Host- oder awsvpc-Modus. Weitere Beispiele dafür, wie Sie den CloudWatch Agenten auf Amazon ECS konfigurieren können, finden Sie im Github-Beispiel-Repository

Im Folgenden finden Sie ein Beispiel für den Bridge-Modus. Wenn das Bridge-Modus-Netzwerk aktiviert ist, muss der Agent mithilfe des `links`-Parameters mit Ihrer Anwendung verknüpft und mithilfe des Containernamens adressiert werden.

```
{
  "containerDefinitions": [
          {
              "name": "webapp",
              "links": [ "cwagent" ],
              "image": "my-org/web-app:latest",
              "memory": 256,
              "cpu": 256,
              "environment": [{
                "name": "AWS_EMF_AGENT_ENDPOINT",
                "value": "tcp://cwagent:25888"
              }],
          },
          {
              "name": "cwagent",
              "mountPoints": [],
              "image": "public.ecr.aws/cloudwatch-agent/cloudwatch-agent:latest",
              "memory": 256,
              "cpu": 256,
              "portMappings": [{
                  "protocol": "tcp",
                  "containerPort": 25888
              }],
              "environment": [{
                "name": "CW_CONFIG_CONTENT",
                "valueFrom": "cwagentconfig"
              }],
          }
      ],
}
```

Im Folgenden finden Sie ein Beispiel für den Host-Modus oder den awsvpc-Modus. Beim Ausführen auf diesen Netzwerkmodi kann der Agent über `localhost` angesprochen werden.

```
{
  "containerDefinitions": [
          {
              "name": "webapp",
```

```
            "image": "my-org/web-app:latest",
            "memory": 256,
            "cpu": 256,
            "environment": [{
              "name": "AWS_EMF_AGENT_ENDPOINT",
              "value": "tcp://127.0.0.1:25888"
            }],
        },
        {
            "name": "cwagent",
            "mountPoints": [],
            "image": "public.ecr.aws/cloudwatch-agent/cloudwatch-agent:latest",
            "memory": 256,
            "cpu": 256,
            "portMappings": [{
                "protocol": "tcp",
                "containerPort": 25888
            }],
            "environment": [{
              "name": "CW_CONFIG_CONTENT",
              "valueFrom": "cwagentconfig"
            }],
        }
    ],
}
```

Note

Im awsvpc-Modus müssen Sie der VPC entweder eine öffentliche IP-Adresse geben (nur Fargate), ein NAT-Gateway einrichten oder einen Logs-VPC-Endpunkt einrichten. CloudWatch Weitere Informationen zum Einrichten einer NAT finden Sie unter NAT-Gateways. Weitere Informationen zum Einrichten eines CloudWatch Logs-VPC-Endpoints finden Sie unter Using CloudWatch Logs with Interface VPC Endpoints.
Im Folgenden finden Sie ein Beispiel dafür, wie Sie einer Aufgabe, die den Fargate-Starttyp verwendet, eine öffentliche IP-Adresse zuweisen.

```
aws ecs run-task \
--cluster {{cluster-name}} \
--task-definition cwagent-fargate \
--region {{region}} \
--launch-type FARGATE \
```

```
--network-configuration
 "awsvpcConfiguration={subnets=[{{subnetId}}],securityGroups=[{{sgId}}],assignPublicIp=ENA
```

Sicherstellen der Berechtigungen

Stellen Sie sicher, dass die IAM-Rolle, die Ihre Aufgaben ausführt, über die Berechtigung zum Lesen aus dem SSM-Parameterspeicher verfügt. Sie können diese Berechtigung hinzufügen, indem Sie die AmazonSSM-Richtlinie anhängen. ReadOnlyAccess Geben Sie dazu den folgenden Befehl ein.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/AmazonSSMReadOnlyAccess
 \
--role-name CWAgentECSExecutionRole
```

Installation des CloudWatch Agenten auf Amazon EKS

Teile dieses Vorgangs können übersprungen werden, wenn Sie CloudWatch Container Insights bereits auf diesem Cluster installiert haben.

Berechtigungen

Wenn Sie Container Insights noch nicht installiert haben, stellen Sie zunächst sicher, dass Ihre Amazon-EKS-Knoten über die entsprechenden IAM-Berechtigungen verfügen. Sie sollten die CloudWatchAgentServerPolicyangehängt haben. Weitere Informationen finden Sie unter Überprüfen Sie die -Voraussetzungen..

Erschaffen ConfigMap

Erstellen Sie eine ConfigMap für den Agenten. Der weist den Agenten ConfigMap außerdem an, einen TCP- oder UDP-Port abzuhören. Verwenden Sie Folgendes ConfigMap.

```
# cwagent-emf-configmap.yaml
apiVersion: v1
data:
  # Any changes here must not break the JSON format
  cwagentconfig.json: |
    {
      "agent": {
        "omit_hostname": true
      },
      "logs": {
```

```
        "metrics_collected": {
          "emf": { }
        }
      }
    }
kind: ConfigMap
metadata:
  name: cwagentemfconfig
  namespace: default
```

Wenn Sie Container Insights bereits installiert haben, fügen Sie Ihrer vorhandenen `"emf": { }` Zeile die folgende Zeile hinzu ConfigMap.

Wenden Sie das an ConfigMap

Geben Sie den folgenden Befehl ein, um das anzuwenden ConfigMap.

```
kubectl apply -f cwagent-emf-configmap.yaml
```

Bereitstellen des Agenten

Um den CloudWatch Agenten als Sidecar bereitzustellen, fügen Sie den Agenten wie im folgenden Beispiel zu Ihrer Pod-Definition hinzu.

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp
  namespace: default
spec:
  containers:
    # Your container definitions go here
    - name: web-app
      image: my-org/web-app:latest
    # CloudWatch Agent configuration
    - name: cloudwatch-agent
      image: public.ecr.aws/cloudwatch-agent/cloudwatch-agent:latest
      imagePullPolicy: Always
      resources:
        limits:
          cpu: 200m
          memory: 100Mi
        requests:
```

```
          cpu: 200m
          memory: 100Mi
      volumeMounts:
        - name: cwagentconfig
          mountPath: /etc/cwagentconfig
      ports:
  # this should match the port configured in the ConfigMap
        - protocol: TCP
          hostPort: 25888
          containerPort: 25888
  volumes:
    - name: cwagentconfig
      configMap:
        name: cwagentemfconfig
```

Verwenden des CloudWatch Agenten zum Senden eingebetteter Logs im metrischen Format

Wenn der CloudWatch Agent installiert ist und ausgeführt wird, können Sie die eingebetteten Protokolle im Metrikformat über TCP oder UDP senden. Beim Senden der Protokolle über den Agenten gibt es zwei Anforderungen:

- Die Protokolle müssen einen `LogGroupName`-Schlüssel enthalten, der dem Agenten mitteilt, welche Protokollgruppe verwendet werden soll.
- Jedes Protokollereignis muss sich in einer einzigen Zeile befinden. Mit anderen Worten, ein Protokollereignis darf das Zeilenumbruchzeichen (\n) nicht enthalten.

Die Protokollereignisse müssen auch der Spezifikation für eingebettete Metrikformate entsprechen. Weitere Informationen finden Sie unter Spezifikation: Eingebettetes Metrikformat .

Wenn Sie Alarme für Metriken erstellen möchten, die im eingebetteten Metrikformat erstellt wurden, siehe Alarme für Metriken setzen, die mit dem eingebetteten Metrikformat erstellt wurden für Empfehlungen.

Im Folgenden finden Sie ein Beispiel für das manuelle Senden von Protokollereignissen aus einer Linux-Bash-Shell. Sie können stattdessen die UDP-Socket-Schnittstellen verwenden, die von Ihrer gewünschten Programmiersprache bereitgestellt werden.

```
echo '{"_aws":{"Timestamp":1574109732004,"LogGroupName":"Foo","CloudWatchMetrics":
[{"Namespace":"MyApp","Dimensions":[["Operation"]],"Metrics":
```

```
[{"Name":"ProcessingLatency","Unit":"Milliseconds","StorageResolution":60}]}]},"Operation":"Agg
 \
> /dev/udp/0.0.0.0/25888
```

> **Note**
>
> Mit dem eingebetteten Metrikformat können Sie die Verarbeitung Ihrer EMF-Protokolle nach Metriken verfolgen, die im `AWS/Logs`-Namespace Ihres Kontos veröffentlicht werden. Diese können verwendet werden, um die fehlgeschlagene Metrikgenerierung von EMF nachzuverfolgen und festzustellen, ob Fehler aufgrund der Analyse oder der Validierung auftreten. Weitere Informationen finden Sie unter Überwachung mit CloudWatch Metriken.

Verwenden des eingebetteten metrischen Formats mit AWS Distro für OpenTelemetry

Sie können das eingebettete metrische Format als Teil des OpenTelemetry Projekts verwenden. OpenTelemetry ist eine Open-Source-Initiative, die Grenzen und Einschränkungen zwischen herstellerspezifischen Formaten für Tracing, Logs und Metriken aufhebt, indem sie einen einzigen Satz von Spezifikationen und APIs anbietet. Weitere Informationen finden Sie unter. OpenTelemetry

Für die Verwendung des eingebetteten metrischen Formats mit OpenTelemetry sind zwei Komponenten erforderlich: eine OpenTelemetry -konforme Datenquelle und AWS Distro for OpenTelemetry Collector, die für die Verwendung mit CloudWatch eingebetteten Protokollen im metrischen Format aktiviert ist.

Wir haben Neuverteilungen der OpenTelemetry Komponenten vorkonfiguriert, die von verwaltet werden, um das Onboarding so AWS einfach wie möglich zu gestalten. Weitere Informationen zur Verwendung des OpenTelemetry eingebetteten metrischen Formats sowie zu anderen AWS Diensten finden Sie unter Distro for.AWS OpenTelemetry

Weitere Informationen zur Sprachunterstützung und -verwendung finden Sie unter AWS Beobachtbarkeit auf Github.

Anzeigen Ihrer Metriken und Protokolle in der Konsole

Nachdem Sie Logs im eingebetteten Metrikformat generiert haben, die Metriken extrahieren, können Sie die Messwerte in der CloudWatch Konsole anzeigen. Eingebettete Metriken weisen

die Dimensionen auf, die Sie beim Generieren der Protokolle angegeben haben. Außerdem weisen eingebettete Metriken, die Sie mit den Clientbibliotheken generiert haben, die folgenden Standarddimensionen auf:

- ServiceType
- ServiceName
- LogGroup

So zeigen Sie Metriken an, die aus Protokollen im eingebetteten Metrikformat generiert wurden

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie einen Namespace aus, den Sie bei der Generierung Ihrer eingebetteten Metriken angegeben haben. Wenn Sie die Clientbibliotheken zum Generieren der Metriken verwendet und keinen Namespace angegeben haben, wählen Sie aws-embedded-metrics. Dies ist der Standardnamespace für eingebettete Metriken, die mithilfe der Clientbibliotheken generiert werden.
4. Wählen Sie eine Metrikdimension aus (z. B. ServiceName).
5. Die Registerkarte All metrics zeigt alle Metriken für diese Dimension im Namespace an. Sie haben die folgenden Möglichkeiten:

    a. Um die Tabelle sortieren, verwenden Sie die Spaltenüberschrift.

    b. Um eine Metrik grafisch darzustellen, müssen Sie das Kontrollkästchen neben der Metrik aktivieren. Um alle Metriken auszuwählen, aktivieren Sie das Kontrollkästchen in der Kopfzeile der Tabelle.

    c. Um nach Ressource zu filtern, müssen Sie zunächst die Ressourcen-ID und dann die Option Zu Suche hinzufügen auswählen.

    d. Um nach Metrik zu filtern, müssen Sie den Metriknamen und anschließend Add to search (Zur Suche hinzufügen) auswählen.

Logs mithilfe von Logs Insights CloudWatch abfragen

Sie können die detaillierten Protokollereignisse im Zusammenhang mit den extrahierten Metriken abfragen, indem Sie CloudWatch Logs Insights verwenden, um tiefe Einblicke in die Hauptursachen von Betriebsereignissen zu erhalten. Einer der Vorteile des Extrahierens von Metriken aus Ihren

Protokollen besteht darin, dass Sie Ihre Protokolle später nach der eindeutigen Metrik (Metrikname plus eindeutiger Dimensionssatz) und Metrikwerten filtern können, um Kontext zu den Ereignissen zu erhalten, die zum aggregierten Metrikwert beigetragen haben.

Um beispielsweise eine betroffene Anfrage-ID oder eine X-Ray-Trace-ID abzurufen, könnten Sie die folgende Abfrage in CloudWatch Logs Insights ausführen.

```
filter Latency > 1000 and Operation = "Aggregator"
| fields RequestId, TraceId
```

Sie können auch Abfragezeitaggregation für Schlüssel mit hoher Kardinalität durchführen, z. B. das Finden der Kunden, die von einem Ereignis betroffen sind. Das folgende Beispiel illustriert dies.

```
filter Latency > 1000 and Operation = "Aggregator"
| stats count() by CustomerId
```

Weitere Informationen finden Sie unter Analysieren von Protokolldaten mit CloudWatch Logs Insights

Alarme für Metriken setzen, die mit dem eingebetteten Metrikformat erstellt wurden

Im Allgemeinen folgt das Erstellen von Alarmen für Metriken, die durch das eingebettete Metrikformat erzeugt wurden, demselben Muster wie das Erstellen von Alarmen für andere Metriken. Weitere Informationen finden Sie unter CloudWatch Amazon-Alarme verwenden.

Die Generierung von Metriken im eingebetteten Metrikformat hängt von Ihrem Ablauf bei der Protokollveröffentlichung ab, da die CloudWatch Protokolle von Logs verarbeitet werden müssen, um in Metriken umgewandelt zu werden. Es ist also wichtig, dass Sie die Protokolle rechtzeitig veröffentlichen, damit Ihre metrischen Datenpunkte innerhalb des Zeitraums erstellt werden, in dem die Alarme ausgewertet werden.

Wenn Sie beabsichtigen, das eingebettete Metrikformat zu verwenden, um hochauflösende Metriken zu senden und Alarme für diese Metriken zu erstellen, empfehlen wir, die CloudWatch Protokolle in einem Intervall von 5 Sekunden oder weniger in Logs zu speichern, um zusätzliche Verzögerungen zu vermeiden, die bei unvollständigen oder fehlenden Daten zu Alarmen führen können. Wenn Sie den CloudWatch Agenten verwenden, können Sie das Löschintervall anpassen, indem Sie den `force_flush_interval` Parameter in der CloudWatch Agentenkonfigurationsdatei festlegen. Dieser Wert ist standardmäßig auf 5 Sekunden eingestellt.

Wenn Sie Lambda auf anderen Plattformen verwenden, bei denen Sie das Intervall für den Protokoll-Flush nicht kontrollieren können, sollten Sie die Verwendung von „M aus N“-Alarmen in Betracht ziehen, um die Anzahl der Datenpunkte zu kontrollieren, die für den Alarm verwendet werden. Weitere Informationen finden Sie unter Auswerten eines Alarms.

AWS Dienste, die CloudWatch Metriken veröffentlichen

Die folgenden AWS Dienste veröffentlichen Metriken auf CloudWatch. Informationen zu den Metriken und Dimensionen finden Sie in der angegebenen Dokumentation.

Service	Namespace	Dokumentation
AWS Amplify	`AWS/AmplifyHosting`	Überwachung
Amazon API Gateway	`AWS/ApiGateway`	Überwachen Sie die API-Ausführung mit Amazon CloudWatch
Amazon AppFlow	`AWS/AppFlow`	Amazon AppFlow mit Amazon überwachen CloudWatch
AWS Service zur Anwendungsmigration	`AWS/MGN`	Überwachung des Anwendungsmigrationsdienstes mit Amazon CloudWatch
AWS App Runner	`AWS/AppRunner`	App Runner-Servicemetriken anzeigen, an die gemeldet wurden CloudWatch
AppStream 2.0	`AWS/AppStream`	Überwachung von Amazon AppStream 2.0-Ressourcen
AWS AppSync	`AWS/AppSync`	CloudWatch Metriken
Amazon Athena	`AWS/Athena`	Überwachung von Athena-Abfragen mit Metriken CloudWatch
Amazon Aurora	`AWS/RDS`	Amazon Aurora-Metriken
AWS Backup	`AWS/Backup`	Überwachung von AWS Backup-Metriken mit CloudWatch
Amazon Bedrock	`AWS/Bedrock`	Überwachung von Amazon Bedrock mit Amazon CloudWatch

Service	Namespace	Dokumentation
AWS Billing and Cost Management	`AWS/Billing`	Überwachung von Gebühren mit Warnungen und Benachrichtigungen
Amazon Braket	`AWS/Braket/By Device`	Überwachung von Amazon Braket mit Amazon CloudWatch
AWS Certificate Manager	`AWS/CertificateManager`	Unterstützte Metriken CloudWatch
AWS Private CA	`AWS/ACMPrivateCA`	Unterstützte CloudWatch Metriken
AWS Chatbot	`AWS/Chatbot`	Überwachung AWS Chatbot mit Amazon CloudWatch
Amazon Chime	`AWS/Chime VoiceConnector`	Überwachung Amazon Chime mit Amazon CloudWatch
Amazon Chime SDK	`AWS/Chime SDK`	Servicemetriken
AWS Client VPN	`AWS/ClientVPN`	Überwachung mit Amazon CloudWatch
Amazon CloudFront	`AWS/CloudFront`	CloudFront Aktivität überwachen mit CloudWatch
AWS CloudHSM	`AWS/CloudHSM`	CloudWatch Metriken abrufen
Amazon CloudSearch	`AWS/CloudSearch`	Überwachung einer CloudSearch Amazon-Domain mit Amazon CloudWatch
AWS CloudTrail	`AWS/CloudTrail`	Unterstützte CloudWatch Metriken

Service	Namespace	Dokumentation
CloudWatch Agent	`CWAgent` oder ein benutzerdefinierter Namespace	Vom CloudWatch Agenten gesammelte Metriken
CloudWatch metrische Streams	`AWS/CloudWatch/MetricStreams`	Überwachen Sie Ihre metrischen Streams mit CloudWatch Metriken
CloudWatch RUM	`AWS/RUM`	CloudWatch Metriken, die Sie mit CloudWatch RUM sammeln können
CloudWatch Synthetics	`CloudWatchSynthetics`	CloudWatch Von den Kanaren veröffentlichte Metriken
CloudWatch Amazon-Protokolle	`AWS/Logs`	Überwachung der Nutzung mit CloudWatch Metriken
AWS CodeBuild	`AWS/CodeBuild`	Überwachung AWS CodeBuild
CodeGuru Amazon-Rezensent		CodeGuru Überprüfer mit Amazon überwachen CloudWatch
Amazon Kendra		Überwachung von Amazon Kendra mit Amazon CloudWatch
Amazon CodeWhisperer	`AWS/CodeWhisperer`	Überwachung Amazon CodeWhisperer mit Amazon CloudWatch
Amazon Cognito	`AWS/Cognito`	Überwachung von Amazon Cognito
Amazon Comprehend	`AWS/Comprehend`	Überwachung von Amazon Comprehend Endpunkten

Service	Namespace	Dokumentation
AWS Config	`AWS/Config`	AWS Config Nutzungs- und Erfolgsmetriken
Amazon Connect	`AWS/Connect`	Überwachung von Amazon Connect in Amazon CloudWatch Metrics
Amazon Data Lifecycle Manager	`AWS/DataLifecycleManager`	Überwachen Sie Ihre Richtlinien mit Amazon CloudWatch
AWS DataSync	`AWS/DataSync`	Überwachung Ihrer Aufgabe
Amazon DataZone		Amazon DataZone mit Amazon überwachen CloudWatch
Amazon DevOps Guru	`AWS/DevOps-Guru`	Überwachung Amazon DevOps Guru mit Amazon CloudWatch
AWS Database Migration Service	`AWS/DMS`	AWS DMS Aufgaben überwachen
AWS Direct Connect	`AWS/DX`	Überwachung mit Amazon CloudWatch
AWS Directory Service	`AWS/DirectoryService`	Ermitteln Sie anhand von CloudWatch Amazon-Metriken, wann Domain-Controller hinzugefügt werden müssen
Amazon DocumentDB	`AWS/DocDB`	Amazon DocumentDB-Metriken
Amazon-DynamoDB	`AWS/DynamoDB`	DynamoDB-Metriken und -Dimensionen

Service	Namespace	Dokumentation
DynamoDB Accelerator (DAX).	`AWS/DAX`	Anzeigen von DAX-Metriken und -Dimensionen
Amazon EC2	`AWS/EC2`	Überwachen Sie Ihre Instances mit CloudWatch
Amazon EC2 Elastic Graphics	`AWS/ElasticGPUs`	Verwendung von CloudWatch Metriken zur Überwachung von Elastic Graphics
Amazon EC2-Spot-Flotte	`AWS/EC2Spot`	CloudWatch Metriken für Spot Fleet
Amazon EC2 Auto Scaling	`AWS/AutoScaling`	Überwachen Sie Ihre Auto Scaling Scaling-Gruppen und -Instances mit CloudWatch
AWS Elastic Beanstalk	`AWS/ElasticBeanstalk`	Veröffentlichen CloudWatch benutzerdefinierter Amazon-Metriken für eine Umgebung
Amazon Elastic Block Store	`AWS/EBS`	CloudWatch Amazon-Metriken für Amazon EBS
Amazon Elastic Container Registry	`AWS/ECR`	Amazon-ECR-Repository-Metriken
Amazon Elastic Container Service	`AWS/ECS`	Amazon CloudWatch ECS-Metriken
Amazon ECS über CloudWatch Container Insights	`ECS/ContainerInsights`	Metriken von Amazon ECS Container Insights

Service	Namespace	Dokumentation
Auto Scaling für Amazon-ECS-Cluster	`AWS/ECS/ManagedScaling`	Auto Scaling für Amazon-ECS-Cluster
AWS Elastic Disaster Recovery		CloudWatch Metriken für DRS
Amazon Elastic File System	`AWS/EFS`	Überwachung mit CloudWatch
Amazon Elastic Inference	`AWS/ElasticInference`	Verwendung von CloudWatch Metriken zur Überwachung von Amazon Elastic Inference
Amazon EKS über CloudWatch Container Insights	`ContainerInsights`	Container-Insights-Metriken für Amazon EKS und Kubernetes
Elastic Load Balancing	`AWS/ApplicationELB`	CloudWatch Metriken für Ihren Application Load Balancer
Elastic Load Balancing	`AWS/NetworkELB`	CloudWatch Metriken für Ihren Network Load Balancer
Elastic Load Balancing	`AWS/GatewayELB`	CloudWatch Metriken für Ihren Gateway Load Balancer
Elastic Load Balancing	`AWS/ELB`	CloudWatch Metriken für Ihren Classic Load Balancer
Amazon Elastic Transcoder	`AWS/ElasticTranscoder`	Überwachung mit Amazon CloudWatch

Service	Namespace	Dokumentation
Amazon ElastiCache für Memcached	`AWS/ElastiCache`	Überwachung der Nutzung mit Metriken CloudWatch
Amazon ElastiCache für Redis	`AWS/ElastiCache`	Überwachung der Nutzung mit Metriken CloudWatch
OpenSearch Amazon-Dienst	`AWS/ES`	Überwachung von OpenSearch Cluster-Metriken mit Amazon CloudWatch
Amazon EMR	`AWS/ElasticMapReduce`	Überwachen Sie Metriken mit CloudWatch
AWS Elemental MediaConnect	`AWS/MediaConnect`	Überwachung MediaConnect mit Amazon CloudWatch
AWS Elemental MediaConvert	`AWS/MediaConvert`	Verwenden von CloudWatch Metriken zum Anzeigen von Metriken für AWS Elemental MediaConvert Ressourcen
AWS Elemental MediaLive	`AWS/MediaLive`	Aktivitäten anhand von CloudWatch Amazon-Metriken überwachen
AWS Elemental MediaPackage	`AWS/MediaPackage`	Überwachung AWS Elemental MediaPackage mit Amazon CloudWatch Metrics
AWS Elemental MediaStore	`AWS/MediaStore`	Überwachung AWS Elemental MediaStore mit Amazon CloudWatch Metrics
AWS Elemental MediaTailor	`AWS/MediaTailor`	Überwachung AWS Elemental MediaTailor mit Amazon CloudWatch
Amazon EventBridge	`AWS/Events`	Überwachung von Amazon EventBridge

Service	Namespace	Dokumentation
Amazon FinSpace		Protokollieren und überwachen
Amazon Forecast		CloudWatch Metriken für Amazon Forecast
Amazon Fraud Detector		Überwachung von Amazon Fraud Detector mit Amazon CloudWatch
Amazon FSx für Lustre	`AWS/FSx`	Überwachung von Amazon FSx for Lustre
Amazon FSx für OpenZFS	`AWS/FSx`	Überwachung mit Amazon CloudWatch
Amazon FSx für Windows File Server	`AWS/FSx`	Überwachung von Amazon FSx for Windows File Server
Amazon FSx für ONTAP NetApp	`AWS/FSx`	Überwachung mit Amazon CloudWatch
Amazon FSx für OpenZFS	`AWS/FSx`	Überwachung mit Amazon CloudWatch
Amazon GameLift	`AWS/GameLift`	Überwachen Sie Amazon GameLift mit CloudWatch
AWS Global Accelerator	`AWS/GlobalAccelerator`	Amazon verwenden CloudWatch mit AWS Global Accelerator
AWS Glue	`Glue`	Überwachung AWS Glue mithilfe von CloudWatch Metriken
AWS Ground Station	`AWS/GroundStation`	Metriken mit Amazon CloudWatch

Service	Namespace	Dokumentation
AWS HealthLake	`AWS/HealthLake`	Überwachung HealthLake mit CloudWatch
Amazon Inspector	`AWS/Inspector`	Überwachung von Amazon Inspector mithilfe CloudWatch
Amazon Interactive Video Service	`AWS/IVS`	Überwachung von Amazon IVS mit Amazon CloudWatch
Amazon Interactive Video Service Chat	`AWS/IVSChat`	Überwachung von Amazon IVS mit Amazon CloudWatch
AWS IoT	`AWS/IoT`	AWS IoT Metriken und Dimensionen
AWS IoT Analytics	`AWS/IoTAnalytics`	Namespace, Metriken und Dimensionen
AWS IoT FleetWise	`AWS/IoTFleetWise`	Überwachung des AWS IoT FleetWise mit Amazon CloudWatch
AWS IoT SiteWise	`AWS/IoTSiteWise`	Überwachung AWS IoT SiteWise mit CloudWatch Amazon-Metriken
AWS IoT TwinMaker	`AWS/IoTTwinMaker`	Überwachung AWS IoT TwinMaker mit CloudWatch Amazon-Metriken
AWS IoT 1-Klick		Überwachung von AWS IoT 1-Click mit Amazon CloudWatch
AWS Key Management Service	`AWS/KMS`	Überwachung mit CloudWatch

Service	Namespace	Dokumentation
Amazon Keyspaces (für Apache Cassandra)	`AWS/Cassandra`	Amazon Keyspaces-Metriken und -Dimensionen
Amazon Kendra		Überwachung von Amazon Kendra mit Amazon CloudWatch
Amazon Managed Service für Apache Flink	`AWS/KinesisAnalytics`	Managed Service für Apache Flink für SQL-Anwendungen: Überwachung mit CloudWatch Managed Service für Apache Flink für Apache Flink: Anzeigen von Metriken und Dimensionen von Amazon Managed Service für Apache Flink
Amazon Data Firehose	`AWS/Firehose`	Überwachung von Firehose mithilfe von Metriken CloudWatch
Amazon-Kinesis-Data-Streams	`AWS/Kinesis`	Überwachung von Amazon Kinesis Data Streams mit Amazon CloudWatch
Amazon Kinesis Video Streams	`AWS/KinesisVideo`	Überwachen von Kinesis Video Streams Streams-Metriken mit CloudWatch
AWS Lambda	`AWS/Lambda`	AWS Lambda Metriken
Amazon Lex	`AWS/Lex`	Überwachung von Amazon Lex mit Amazon CloudWatch

AWS License Manager	`AWSLicenseManager/licenseUsage`
	
	`AWS/LicenseManager/LinuxSubscriptions`
		Überwachung der Lizenznutzung mit Amazon CloudWatch
		
		Nutzungsmetriken und CloudWatch Amazon-Alarme für Linux-Abonnements

Service	Namespace	Dokumentation
Amazon Location Service	`AWS/Location`	Nach Amazon exportierte Amazon Location Service Service-Metriken CloudWatch
Amazon Lookout für Equipment	`AWS/lookoutequipment`	Überwachung von Lookout for Equipment mit Amazon CloudWatch
Amazon Lookout für Metrics	`AWS/LookoutMetrics`	Überwachung von Lookout for Metrics mit Amazon CloudWatch
Amazon Lookout für Vision	`AWS/LookoutVision`	Überwachung von Lookout for Vision mit Amazon CloudWatch
AWS Mainframe-Modernisierung		Überwachung der AWS Mainframe-Modernisierung mit Amazon CloudWatch
Amazon Machine Learning	`AWS/ML`	Überwachung von Amazon ML mit CloudWatch Metriken
Amazon Managed Blockchain	`AWS/managedblockchain`	Hyperledger Fabric Peer Node Metrics auf Amazon Managed Blockchain verwenden
Amazon Managed Service für Prometheus	`AWS/Prometheus`	CloudWatch Amazon-Metriken
Amazon Managed Streaming für Apache Kafka	`AWS/Kafka`	Überwachung von Amazon MSK mit Amazon CloudWatch

Service	Namespace	Dokumentation
Amazon Managed Streaming für Apache Kafka	`AWS/Kafka Connect`	Überwachen von MSK Connect
Amazon Managed Workflows für Apache Airflow	`AWS/MWAA`	Container-, Warteschlangen- und Datenbankmetriken für Amazon MWAA
Amazon MemoryDB for Redis	`AWS/MemoryDB`	Metriken überwachen CloudWatch
Amazon MQ	`AWS/AmazonMQ`	Überwachung von Amazon MQ-Brokern mithilfe von Amazon CloudWatch
Amazon Neptune	`AWS/Neptune`	Überwachung von Neptune mit CloudWatch
AWS Network Firewall	`AWS/NetworkFirewall`	AWS Network Firewall Metriken in Amazon CloudWatch
AWS Netzwerkmanager	`AWS/NetworkManager`	CloudWatch Metriken für lokale Ressourcen
Amazon Nimble Studio	`AWS/NimbleStudio`	Nimble Studio mit Amazon überwachen CloudWatch
AWS HealthOmics	`AWS/Omics`	Überwachung AWS HealthOmics mit Amazon CloudWatch
AWS OpsWorks	`AWS/OpsWorks`	Stacks mit Amazon überwachen CloudWatch
AWS Outposts	`AWS/Outposts`	CloudWatch Metriken für AWS Outposts

Service	Namespace	Dokumentation
AWS Panorama	`AWS/PanoramaDeviceMetrics`	Überwachung von Appliances und Anwendungen mit Amazon CloudWatch
Amazon Personalize	`AWS/Personalize`	CloudWatch Metriken für Amazon Personalize
Amazon Pinpoint	`AWS/Pinpoint`	Amazon Pinpoint Metriken anzeigen in CloudWatch
Amazon Polly	`AWS/Polly`	Integration CloudWatch mit Amazon Polly
AWS PrivateLink	`AWS/PrivateLinkEndpoints`	CloudWatch Metriken für AWS PrivateLink
AWS PrivateLink	`AWS/PrivateLinkServices`	CloudWatch Metriken für AWS PrivateLink
AWS Privates 5G	`AWS/Private5G`	CloudWatch Amazon-Metriken
Amazon QLDB	`AWS/QLDB`	Daten in Amazon überwachen QuickSight
Amazon QuickSight	`AWS/QuickSight`	Überwachung mit Amazon CloudWatch
Amazon-Redshift	`AWS/Redshift`	Leistungsdaten von Amazon Redshift
Amazon Relational Database Service	`AWS/RDS`	Überwachung von Amazon RDS-Metriken mit Amazon CloudWatch

Service	Namespace	Dokumentation
Amazon Rekognition	`AWS/Rekognition`	Überwachung von Rekognition mit Amazon CloudWatch
AWS re:Post Privat	`AWS/rePostPrivate`	AWS re:Post Private Überwachung mit Amazon CloudWatch
AWS RoboMaker	`AWS/Robomaker`	Überwachung AWS RoboMaker mit Amazon CloudWatch
Amazon Route 53	`AWS/Route53`	Überwachung von Amazon Route 53
Route 53 Application Recovery-Controller	`AWS/Route53RecoveryReadiness`	Amazon CloudWatch mit Application Recovery Controller verwenden
Amazon SageMaker	`AWS/SageMaker`	Überwachung SageMaker mit CloudWatch
SageMaker Amazon-Modellbau-Pipelines	`AWS/SageMaker/ModelBuildingPipeline`	SageMaker Metriken für Pipelines
AWS Secrets Manager	`AWS/SecretsManager`	Überwachung von Secrets Manager mit Amazon CloudWatch
Amazon Security Lake	`AWS/SecurityLake`	CloudWatch Metriken für Amazon Security Lake
Servicekatalog	`AWS/ServiceCatalog`	CloudWatch Kennzahlen Service Catalog
AWS Shield Advanced	`AWS/DDoSProtection`	Überwachung mit CloudWatch

Service	Namespace	Dokumentation
Amazon Simple Email Service	AWS/SES	Amazon SES SES-Ereignisdaten werden abgerufen von CloudWatch
AWS SimSpace Weaver	AWS/simspaceweaver	Überwachung AWS SimSpace Weaver mit Amazon CloudWatch
Amazon Simple Notification Service	AWS/SNS	Überwachung von Amazon SNS mit CloudWatch
Amazon Simple Queue Service	AWS/SQS	Überwachen Amazon SQS SQS-Warteschlangen mithilfe CloudWatch
Amazon S3	AWS/S3	Metriken mit Amazon überwachen CloudWatch
S3 Storage Lens	AWS/S3/Storage-Lens	Überwachen Sie die Metriken von S3 Storage Lens in CloudWatch
Amazon Simple Workflow Service	AWS/SWF	Amazon SWF-Metriken für CloudWatch
AWS Step Functions	AWS/States	Step Functions überwachen mit CloudWatch
AWS Storage Gateway	AWS/StorageGateway	Verwenden von CloudWatch Amazon-Metriken
AWS Systems Manager Befehl ausführen	AWS/SSM-RunCommand	Überwachen von Run-Command-Metriken mit CloudWatch
Amazon Textract	AWS/Textract	CloudWatch Metriken für Amazon Textract
Amazon Timestream	AWS/Timestream	Timestream-Metriken und Dimensionen

Service	Namespace	Dokumentation
AWS Transfer for SFTP	`AWS/Transfer`	AWS SFTP CloudWatch Metriken
Amazon Transcribe	`AWS/Transcribe`	Überwachung Amazon Transcribe mit Amazon CloudWatch
Amazon Translate	`AWS/Translate`	CloudWatch Metriken und Dimensionen für Amazon Translate
AWS Trusted Advisor	`AWS/TrustedAdvisor`	Trusted Advisor Advisor-Alarme erstellen mit CloudWatch
Amazon VPC	`AWS/NATGateway`	Überwachen Sie Ihr NAT-Gateway mit CloudWatch
Amazon VPC	`AWS/TransitGateway`	CloudWatch Metriken für Ihre Transit-Gateways
Amazon VPC	`AWS/VPN`	Überwachung mit CloudWatch
Amazon VPC IP Address Manager	`AWS/IPAM`	Alarme mit Amazon erstellen CloudWatch
AWS WAF	`AWS/WAFV2` für AWS WAF Ressourcen `WAF`für AWS WAF klassische Ressourcen	Überwachung mit CloudWatch
Amazon WorkMail	`AWS/WorkMail`	Überwachung Amazon WorkMail mit Amazon CloudWatch
Amazon WorkSpaces	`AWS/WorkSpaces`	Überwachen Sie Ihre WorkSpaces CloudWatch Nutzungsmetriken

Service	Namespace	Dokumentation
Amazon WorkSpaces Web	`AWS/WorkSpacesWeb`	Überwachung von Amazon WorkSpaces Web mit Amazon CloudWatch

AWS Nutzungsmetriken

CloudWatch sammelt Messwerte, die die Nutzung einiger AWS Ressourcen und APIs verfolgen. Diese Metriken werden im `AWS/Usage`-Namespace veröffentlicht. Mit Nutzungsmetriken CloudWatch können Sie die Nutzung proaktiv verwalten, indem Sie Metriken in der CloudWatch Konsole visualisieren, benutzerdefinierte Dashboards erstellen, Änderungen in der Aktivität mithilfe von CloudWatch Anomalieerkennung erkennen und Alarme konfigurieren, die Sie benachrichtigen, wenn sich die Nutzung einem Schwellenwert nähert.

Einige AWS Dienste integrieren diese Nutzungsmetriken mit Service Quotas. Für diese Dienste können Sie die Nutzung CloudWatch Ihrer Servicekontingenten durch Ihr Konto verwalten. Weitere Informationen finden Sie unter Visualisierung Ihrer Service Quotas und Einstellung von Alarmen.

Themen

Visualisierung Ihrer Service Quotas und Einstellung von Alarmen

Bei einigen AWS Diensten können Sie die Nutzungsmetriken verwenden, um Ihre aktuelle Servicenutzung in CloudWatch Diagrammen und Dashboards zu visualisieren. Sie können eine mathematische CloudWatch Metrikfunktion verwenden, um die Servicequotas für diese Ressourcen in Ihren Diagrammen anzuzeigen. Sie können auch Alarme konfigurieren, die Sie warnen, wenn sich Ihre Nutzung einem Service Quotas nähert. Weitere Informationen zu Servicekontingenten finden Sie unter Was sind Service Quotas im Benutzerhandbuch für Service Quotas.

Wenn Sie mit einem Konto angemeldet sind, das als Überwachungskonto in der CloudWatch kontenübergreifenden Observability eingerichtet ist, können Sie dieses Monitoring-Konto verwenden, um Servicequotas zu visualisieren und Alarme für Metriken in den Quellkonten einzustellen, die mit diesem Monitoring-Konto verknüpft sind. Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.

Gegenwärtig integrieren die folgenden Dienste ihre Nutzungsmetriken in Service Quotas:

- AWS CloudHSM
- Amazon Chime SDK

- Amazon CloudWatch
- CloudWatch Amazon-Protokolle
- Amazon-DynamoDB
- Amazon EC2
- Amazon Elastic Container Registry
- Elastic Load Balancing
- AWS Fargate
- AWS Fault Injection Service
- AWS Interaktiver Videodienst
- AWS Key Management Service
- Amazon Data Firehose
- Amazon Location Service
- Amazon Managed Blockchain (AMB) -Abfrage
- AWS RoboMaker
- Amazon SageMaker

So visualisieren Sie ein Service Quotas und legen optional einen Alarm fest

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie auf der Registerkarte Alle Metriken die Option Nutzung und dann Nach AWS Ressource aus.

   Die Liste der Service Quotas-Nutzungsmetriken wird angezeigt.
4. Aktivieren Sie das Kontrollkästchen neben einer der Metriken.

   Das Diagramm zeigt Ihre aktuelle Nutzung dieser AWS Ressource.
5. Gehen Sie wie folgt vor, um Service Quotas in das Diagramm aufzunehmen:

   a. Wählen Sie die Registerkarte Graphed metrics (Grafisch dargestellte Metriken) aus.

   b. Wählen Sie Math expression (Mathematischer Ausdruck), Start with an empty expression (Mit einem leeren Ausdruck beginnen). Geben Sie in der neuen Zeile unter Details **`SERVICE_QUOTA(m1)`** ein.

Dem Diagramm wird eine neue Linie hinzugefügt, die Service Quotas für die in der Metrik dargestellten Ressource anzeigt.

6. Um die aktuelle Nutzung als Prozentsatz des Kontingents anzuzeigen, fügen Sie einen neuen Ausdruck hinzu oder ändern Sie den aktuellen SERVICE_QUOTA-Ausdruck. Der zu verwendende neue Ausdruck lautet „**`m1/SERVICE_QUOTA(m1)*100`**“.
7. (Optional) Gehen Sie wie folgt vor, um einen Alarm festzulegen, der Sie benachrichtigt, wenn Sie sich Service Quotas nähern:

   a. Wählen Sie in der Zeile „**`m1/SERVICE_QUOTA(m1)*100`**“ unter Actions (Aktionen) das Alarmsymbol aus. Es sieht aus wie eine Glocke.

      Die Seite „Alarmerstellung“ wird angezeigt.

   b. Vergewissern Sie sich unter Conditions (Bedingungen), dass der Threshhold type (Schwellenwert-Typ) Static (Statisch) ist und Whenever Expression1 ist auf Greater (Größer) festgelegt ist. Unter als geben Sie **`80`** ein. Dadurch wird ein Alarm ausgelöst, der in den Zustand ALARM übergeht, wenn die Nutzung 80 Prozent des Kontingents überschreitet.

   c. Wählen Sie Next (Weiter).

   d. Auf der nächsten Seite können Sie ein Amazon-SNS-Thema auswählen oder ein neues erstellen und dann wählen Sie Weiter. Das Thema, das Sie auswählen, wird benachrichtigt, wenn der Alarm in den ALARM-Status wechselt.

   e. Geben Sie auf der nächsten Seite einen Namen und eine Beschreibung für den Alarm ein und wählen Sie dann Next (Weiter).

   f. Wählen Sie Create alarm (Alarm erstellen).

AWS Kennzahlen zur API-Nutzung

Die meisten APIs, die die AWS CloudTrail Protokollierung unterstützen, melden auch Nutzungsmetriken an CloudWatch. Mit den API-Nutzungsmetriken CloudWatch können Sie die API-Nutzung proaktiv verwalten, indem Sie Metriken in der CloudWatch Konsole visualisieren, benutzerdefinierte Dashboards erstellen, Änderungen in der Aktivität mit der CloudWatch Anomalieerkennung erkennen und Alarme konfigurieren, die eine Warnung ausgeben, wenn sich die Nutzung einem Schwellenwert nähert.

In der folgenden Tabelle sind die Dienste aufgeführt, an die API-Nutzungsmetriken gemeldet werden CloudWatch, sowie der Wert, der für die `Service` Dimension verwendet werden soll, um die Nutzungsmetriken dieses Dienstes zu sehen.

Service	Der Wert für die **`Service`**-Dimension
AWS Identity and Access Management Access Analyzer	`Access Analyzer`
AWS Account Management	`Account Management`
Alexa for Business	`A4B`
Amazon API Gateway	`API Gateway`
AWS App Mesh	`App Mesh`
AWS AppConfig	`AWS AppConfig`
Amazon AppFlow	`AppFlow`
Application Auto Scaling	`Application Auto Scaling`
Application Discovery Service	`Application Discovery Service`
Amazon AppStream	`AppStream`
AppStream 2.0 Image Builder	`Image Builder`
Amazon Athena	`Athena`
AWS Audit Manager	`Audit Manager`
AWS Backup	`Backup`
AWS Batch	`Batch`
Amazon Braket	`Braket`
AWS Budgets	`Budgets`

Service	Der Wert für die **`Service`**-Dimension
AWS Certificate Manager	`Certificate Manager`
Amazon Chime SDK	`ChimeSDK`
Amazon Cloud Directory	`Cloud Directory`
AWS Cloud Map	`Cloud Map`
AWS CloudFormation	`CloudFormation`
AWS CloudHSM	`CloudHSM`
Amazon CloudSearch	`CloudSearch`
AWS CloudShell	`CloudShell`
AWS CloudTrail	`CloudTrail`
Amazon CloudWatch	`CloudWatch`
CloudWatch Amazon-Protokolle	`Logs`
Einblicke in CloudWatch Amazon-Anwendungen	`CloudWatch Application Insights`
AWS CodeBuild	`CodeBuild`
AWS CodeCommit	`CodeCommit`
Amazon CodeGuru Profiler	`CodeGuru Profiler`
AWS CodePipeline	`CodePipeline`
AWS CodeStar	`CodeStar`
AWS CodeStar Benachrichtigungen	`CodeStar Notifications`
AWS CodeStar Verbindungen	`CodeStar Connections`
Amazon-Cognito-Identitätspools	`Cognito Identity Pools`

Service	Der Wert für die **`Service`**-Dimension
Amazon Cognito Sync	`Cognito Sync`
Amazon Comprehend	`Comprehend`
Amazon Comprehend Medical	`Comprehend Medical`
AWS Compute Optimizer	`ComputeOptimzier`
Amazon Connect	`Connect`
Amazon Connect Customer Profiles	`Customer Profiles`
AWS Kosten- und Nutzungsberichte	`Cost and Usage Report`
AWS Cost Explorer	`Cost Explorer`
AWS Data Exchange	`Data Exchange`
AWS Manager für den Datenlebenszyklus	`Data Lifecycle Manager`
AWS Database Migration Service	`Database Migration Service`
AWS DataSync	`DataSync`
AWS DeepLens	`AWS DeepLens`
Amazon Detective	`Detective`
Device Advisor	`Device Advisor`
AWS Direct Connect	`Direct Connect`
AWS Directory Service	`Directory Service`
DynamoDB Accelerator	`DynamoDBAccelerator`
Amazon EC2	`EC2`
EC2 Auto Scaling	`EC2 Auto Scaling`

Service	Der Wert für die **`Service`**-Dimension
Amazon Elastic Container Registry	`ECR Public`
Amazon Elastic Container Service	`ECS`
Amazon Elastic File System	`EFS`
Amazon Elastic Kubernetes Service	`EKS`
AWS Elastic Beanstalk	`Elastic Beanstalk`
Amazon Elastic Inference	`Elastic Inference`
Elastic Load Balancing	`Elastic Load Balancing`
Amazon EMR	`EMR Containers`
AWS Firewall Manager	`Firewall Manager`
Amazon FSx	`FSx`
Amazon GameLift	`GameLift`
AWS Glue DataBrew	`DataBrew`
Amazon Managed Grafana	`Grafana`
AWS IoT Greengrass	`Greengrass`
AWS Ground Station	`Ground Station`
AWS Health APIs und Benachrichtigungen	`AWS Health APIs And Notifications`
Amazon Interactive Video Service	`IVS`
AWS IoT Core	`IoT`
AWS IoT 1-Klick	`IoT 1-Click`
AWS IoT Events	`IoT Events`

Service	Der Wert für die **`Service`**-Dimension
AWS IoT RoboRunner	`IoT RoboRunner`
AWS IoT SiteWise	`IoT Sitewise`
AWS IoT Wireless	`IoT Wireless`
Amazon Kendra	`Kendra`
Amazon Keyspaces (für Apache Cassandra)	`Keyspaces`
Amazon Managed Service für Apache Flink	`Kinesis Analytics`
Amazon Data Firehose	`Firehose`
Kinesis Video Streams	`Kinesis Video Streams`
AWS Key Management Service	`KMS`
AWS Lambda	`Lambda`
AWS Launch Wizard	`Launch Wizard`
Amazon Lex	`Amazon Lex`
Amazon Lightsail	`Lightsail`
Amazon Location Service	`Location`
Amazon Lookout für Vision	`Lookout for Vision`
Amazon Machine Learning	`Amazon Machine Learning`
Amazon Macie	`Macie`
Amazon Managed Blockchain (AMB) -Abfrage	`Amazon Managed Blockchain Query`
AWS Managed Services	`AWS Managed Services`
AWS Marketplace Commerce Analytics	`Marketplace Analytics Service`

Service	Der Wert für die **`Service`**-Dimension
AWS Elemental MediaConnect	`MediaConnect`
AWS Elemental MediaConvert	`MediaConvert`
AWS Elemental MediaLive	`MediaLive`
AWS Elemental MediaStore	`Mediastore`
AWS Elemental MediaTailor	`MediaTailor`
AWS Mobile Hub	`Mobile Hub`
AWS Network Firewall	`Network Firewall`
AWS OpsWorks	`OpsWorks`
AWS OpsWorks für das Konfigurationsmanagement	`OPsWorks CM`
AWS Outposts	`Outposts`
AWS Organizations	`Organizations`
Amazon RDS Performance Insights	`Performance Insights`
Amazon Pinpoint	`Pinpoint`
AWS Private Certificate Authority	`Private Certificate Authority`
Amazon Managed Service für Prometheus	`Prometheus`
AWS Proton	`Proton`
Amazon Quantum Ledger Database (Amazon QLDB)	`QLDB`
Amazon RDS	`RDS`
Amazon-Redshift	`Redshift Data API`

Service	Der Wert für die **`Service`**-Dimension
Amazon Rekognition	`Rekognition`
AWS Resource Access Manager	`Resource Access Manager`
AWS Resource Groups	`Resource Groups`
AWS Resource Groups Tagging API	`Resource Groups Tagging API`
AWS RoboMaker	`RoboMaker`
Amazon Route 53-Domains	`Route 53 Domains`
Amazon Route 53 Resolver	`Route 53 Resolver`
Amazon S3	`S3`
Amazon S3 Glacier	`Amazon S3 Glacier`
Amazon SageMaker Runtime	`Sagemaker`
Savings Plans	`Savings Plans`
AWS Secrets Manager	`Secrets Manager`
AWS Security Hub	`Security Hub`
AWS Server Migration Service	`AWS Server Migration Service`
AWS Service Catalog AppRegistry	`Service Catalog AppRegistry`
Service Quotas	`Service Quotas`
AWS Shield	`Shield`
AWS Unterzeichner	`Signer`
Amazon Simple Notification Service	`SNS`
Amazon Simple Email Service	`SES`

Service	Der Wert für die **`Service`**-Dimension
Amazon Simple Queue Service	`SQS`
Identity Store	`Identity Store`
Storage Gateway	`Storage Gateway`
AWS Support	`Support`
Amazon Simple Workflow Service	`SWF`
Amazon Textract	`Textract`
AWS IoT Things Graph	`ThingsGraph`
Amazon Timestream	`Timestream`
Amazon Transcribe	`Transcribe`
Amazon Translate	`Translate`
Streaming-Transkription von Amazon Transcribe	`Transcribe Streaming`
AWS Transfer Family	`Transfer`
AWS WAF	`WAF`
Amazon WorkDocs	`Amazon WorkDocs`
Amazon WorkLink	`WorkLink`
Amazon WorkMail	`Amazon WorkMail`
Amazon WorkSpaces	`Workspaces`
AWS X-Ray	`X-Ray`

Einige Services melden Nutzungsmetriken auch für zusätzliche APIs. Um zu sehen, ob eine API Nutzungsmetriken meldet CloudWatch, verwenden Sie die CloudWatch Konsole, um die von diesem Service gemeldeten Metriken im `AWS/Usage` Namespace zu sehen.

Um die Liste der APIs eines Dienstes anzuzeigen, an die Nutzungsmetriken gemeldet werden CloudWatch

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie auf der Registerkarte Alle Metriken die Option Nutzung und dann Nach AWS Ressource aus.
4. Geben Sie im Suchfeld neben der Liste der Metriken den Namen des Services ein. Die Metriken werden nach dem von Ihnen eingegebenen Service gefiltert.

CloudWatch Nutzungsmetriken

CloudWatch sammelt Metriken, die die Nutzung einiger AWS Ressourcen verfolgen. Diese Metriken entsprechen AWS Servicekontingenten. Die Verfolgung dieser Metriken kann Ihnen dabei helfen, Ihre Kontingente proaktiv zu verwalten. Weitere Informationen finden Sie unter Visualisierung Ihrer Service Quotas und Einstellung von Alarmen.

Die Metriken für die Service-Kontingentnutzung befinden sich im `AWS/Usage`-Namespace und werden jede Minute erfasst.

Derzeit ist der einzige Metrikname in diesem Namespace, der CloudWatch veröffentlicht,. `CallCount` Diese Metrik wird mit den Dimensionen `Resource`, `Service`, und `Type` veröffentlicht. Die `Resource`-Dimension gibt den Namen der nachverfolgten API-Operation an. Zum Beispiel die `CallCount` Metrik mit den Dimensionen `"Type": "API"` und `"Resource": "PutMetricData"` gibt an`"Service": "CloudWatch"`, wie oft der CloudWatch `PutMetricData` API-Vorgang in Ihrem Konto aufgerufen wurde.

Die `CallCount`-Metrik hat keine angegebene Einheit. Die nützlichste Statistik für die Metrik ist `SUM`, die die Gesamtanzahl der Operationen für den 1-Minuten-Zeitraum darstellt.

Metriken

Metrik	Beschreibung
`CallCount`	Die Anzahl der angegebenen Operationen, die in Ihrem Konto ausgeführt werden.

Dimensions (Abmessungen)

Dimension	Beschreibung
`Service`	Der Name des AWS Dienstes, der die Ressource enthält. Für CloudWatch Nutzungsmetriken lautet der Wert für diese Dimension`CloudWatch` .
`Class`	Die Klasse der Ressource, die verfolgt wird. CloudWatch API-Nutzungsmetriken verwenden diese Dimension mit einem Wert von`None`.
`Type`	Der Typ der nachverfolgten Ressource. Wenn die `Service`-Dimension `CloudWatch` ist, ist `API` der derzeit einzige gültige Wert für `Type`.
`Resource`	Der Name der API-Operation. Gültige Werte sind unter anderem: `DeleteAlarms` , `DeleteDashboards` , `DescribeAlarmHistory` , `DescribeAlarms` , `GetDashboard` , `GetMetricData` , `GetMetricStatistics` , `ListMetrics` , `PutDashboard` und `PutMetricData`

CloudWatch Tutorials

Die folgenden Szenarien veranschaulichen die Verwendung von Amazon CloudWatch. Im ersten Szenario verwenden Sie die CloudWatch Konsole, um einen Abrechnungsalarm zu erstellen, der Ihre AWS Nutzung verfolgt und Sie darüber informiert, wenn Sie einen bestimmten Ausgabenschwellenwert überschritten haben. Im zweiten, fortgeschritteneren Szenario verwenden Sie AWS Command Line Interface (AWS CLI), um eine einzelne Metrik für eine hypothetische Anwendung mit dem Namen zu veröffentlichen. GetStarted

Szenarien

- Überwachung Ihrer geschätzten Gebühren
- Veröffentlichung von Metriken

Szenario: Überwachen Sie Ihre geschätzten Gebühren mit CloudWatch

In diesem Szenario erstellen Sie einen CloudWatch Amazon-Alarm, um Ihre geschätzten Gebühren zu überwachen. Wenn Sie die Überwachung der geschätzten Gebühren für Ihr AWS Konto aktivieren, werden die geschätzten Gebühren berechnet und mehrmals täglich CloudWatch als Metrikdaten an sie gesendet.

Die metrischen Fakturierungsdaten werden in der Region USA Ost (Nord-Virginia) gespeichert und stellen die weltweiten Gebühren dar. Zu diesen Daten gehören die geschätzten Gebühren für jeden Dienst, den Sie nutzen, sowie die geschätzte Gesamtsumme Ihrer AWS Gebühren. AWS

Sie können sich Benachrichtigungen per E-Mail zusenden lassen, wenn die Gebühren ein bestimmtes Limit überschreiten. Diese Benachrichtigungen werden durch Amazon Simple Notification Service (Amazon SNS) ausgelöst CloudWatch und Nachrichten gesendet.

> **Note**
>
> Informationen zur Analyse von CloudWatch Gebühren, die Ihnen bereits in Rechnung gestellt wurden, finden Sie unter. CloudWatch Abrechnung und Kosten

Aufgaben

Schritt 1: Gebührenlimit-Warnung aktivieren

Bevor Sie einen Alarm für Ihre geschätzten Gebühren erstellen können, müssen Sie die Fakturierungsbenachrichtigungen aktivieren, damit Sie Ihre geschätzten AWS Gebühren überwachen und anhand von Abrechnungskennzahlen einen Alarm erstellen können. Wenn Sie Gebührenlimit-Warnungen aktiviert haben, können Sie die Datenerfassung nicht deaktivieren. Aber Sie können die von Ihnen erstellten Gebührenlimit-Warnungen löschen.

Wenn Sie Gebührenlimit-Warnungen zum ersten Mal aktivieren, dauert es etwa 15 Minuten, bevor Sie die Gebührendaten anzeigen und Gebührenlimit-Warnungen einrichten können.

Voraussetzungen

- Sie müssen mit den Anmeldedaten eines Root-Benutzers oder eines Benutzers angemeldet sein, der die Berechtigung hat, Fakturierungsinformationen einzusehen.
- Für konsolidierte Fakturierungskonten können die Gebührendaten für die verknüpften Konten durch eine Anmeldung als Zahlungskonto abgerufen werden. Sie können die Gebührendaten für die geschätzten Gesamtkosten und die geschätzten Gebühren nach Service für die einzelnen verknüpften Konten sowie für das konsolidierte Konto anzeigen.
- In einem konsolidierten Fakturierungskonto werden die Metriken für verknüpfte Konten nur erfasst, wenn das Zahlerkonto die Voreinstellung Fakturierungsbenachrichtigungen erhalten aktiviert. Wenn Sie ändern, welches Konto Ihr Management-/Zahler-Konto ist, müssen Sie die Gebührenlimit-Warnung im neuen Management-/Zahler-Konto aktivieren.
- Das Konto darf nicht Teil des Amazon Partner Network (APN) sein, da Abrechnungskennzahlen nicht CloudWatch für APN-Konten veröffentlicht werden. Weitere Informationen finden Sie unter AWS -Partnernetzwerk.

So aktivieren Sie die Überwachung der geschätzten Gebühren

1. Öffnen Sie die AWS Billing Konsole unter https://console.aws.amazon.com/billing/.

2. Wählen Sie im Navigationsbereich die Option Fakturierungseinstellungen aus.
3. Wählen Sie unter Präferenzen für Warnungen die Option Bearbeiten aus.
4. Wählen Sie „ CloudWatch Rechnungsbenachrichtigungen empfangen“.
5. Klicken Sie auf Präferenzen speichern.

Schritt 2: Erstellen eines Abrechnungsalarms

> **Important**
>
> Bevor Sie einen Fakturierungsalarm erstellen, müssen Sie Ihre Region auf USA Ost (Nord-Virginia) setzen. Die metrischen Fakturierungsdaten werden in dieser Region gespeichert und stellen die weltweiten Gebühren dar. Sie müssen Gebührenlimit-Warnungen in Ihrem Konto oder im Management-/Zahler-Konto aktivieren (wenn Sie die konsolidierte Abrechnung verwenden). Weitere Informationen finden Sie unter Schritt 1: Gebührenlimit-Warnung aktivieren.

In diesem Verfahren erstellen Sie einen Alarm, der eine Benachrichtigung sendet, wenn Ihre geschätzten Gebühren einen definierten Schwellenwert AWS überschreiten.

So erstellen Sie mit der CloudWatch Konsole einen Abrechnungsalarm

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Alarme und dann Alle Alarme aus.
3. Wählen Sie Create alarm (Alarm erstellen) aus.
4. Wählen Sie Select metric (Metrik auswählen) aus. Wählen Sie unter Browse (Durchsuchen) die Option Billing (Fakturierung) und dann Estimated Charge (Geschätzte Gesamtgebühr) aus.

   > **Note**
   >
   > Wenn die Metrik Fakturierung/Geschätzte Gesamtgebühr nicht angezeigt wird, aktivieren Sie Fakturierungsalarme und ändern Sie Ihre Region in USA Ost (Nord-Virginia). Weitere Informationen finden Sie unter Aktivieren von Abrechnung-Alarmen.

5. Wählen Sie das Feld für die EstimatedChargesMetrik aus, und wählen Sie dann Metrik auswählen aus.

6. Wählen Sie für Statistic (Statistik) Maximum aus.
7. Wählen Sie als Period (Zeitraum) 6 hours (6 Stunden) aus.
8. Wählen Sie für Threshold type (Schwellenwerttyp) die Option Static (Statisch) aus.
9. Für Wann immer EstimatedCharges es ist. , wählen Sie Größer.
10. Unter als . . . , definieren Sie den Wert, bei dem Ihr Alarm ausgelöst werden soll. Zum Beispiel, **200** USD.

    Die EstimatedChargesmetrischen Werte sind nur in US-Dollar (USD) angegeben, und die Währungsumrechnung wird von Amazon Services LLC bereitgestellt. Weitere Informationen finden Sie unter Was ist AWS Billing? .

    > **Note**
    >
    > Nachdem Sie einen Schwellenwert definiert haben, werden Ihre geschätzten Gebühren für den aktuellen Monat im Vorschaudiagramm angezeigt.

11. Wählen Sie Zusätzliche Konfiguration und führen Sie Folgendes aus:
    - Geben Sie für Datapoints to alarm (zu alarmierende Datenpunkte) 1 out of 1 (1 von 1) an.
    - Wählen Sie für Missing data treatment (Behandlung fehlender Daten) die Option Treat missing data as missing (Fehlende Daten als fehlend behandeln) aus.
12. Wählen Sie Weiter aus.
13. Stellen Sie sicher, dass unter Benachrichtigung die Option Bei Alarm ausgewählt ist. Legen Sie dann ein Amazon-SNS-Thema fest, das benachrichtigt werden soll, wenn sich der Alarm im Status `ALARM` befindet. Das Amazon-SNS-Thema kann Ihre E-Mail-Adresse enthalten, sodass Sie eine E-Mail erhalten, wenn der Rechnungsbetrag den von Ihnen angegebenen Schwellenwert überschreitet.

    Sie können ein vorhandenes Amazon-SNS-Thema auswählen, ein neues Amazon-SNS-Thema erstellen oder einen Themen-ARN verwenden, um ein anderes Konto zu benachrichtigen. Wenn Sie mehrere Benachrichtigungen für den gleichen Alarmstatus oder für verschiedene Alarm-Statuswerte senden möchten, wählen Sie Add notification (Benachrichtigung hinzufügen) aus.
14. Wählen Sie Weiter aus.
15. Geben Sie Name and description (Namen und Beschreibung) für Ihren Alarm ein.
    - (Optional) Geben Sie eine Beschreibung für Ihren Alarm ein.

16. Wählen Sie Weiter aus.
17. Vergewissern Sie sich unter Preview and create (Vorschau anzeigen und erstellen), ob Ihre Konfiguration korrekt ist, und wählen Sie Create alarm (Alarm erstellen) aus.

Schritt 3: Überprüfen des Alarm-Status

Überprüfen Sie nun den Status des Fakturierungsalarms, den Sie gerade erstellt haben.

So überprüfen Sie den Alarmstatus

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Ändern Sie, falls erforderlich, die Region zu USA Ost (Nord-Virginia). Die metrischen Fakturierungsdaten werden in dieser Region gespeichert und stellen die weltweiten Gebühren dar.
3. Klicken Sie im Navigationsbereich auf Alarms (Alarme).
4. Aktivieren Sie das Kontrollkästchen neben dem Alarm. Der Status "Bestätigung ausstehend" wird solange angezeigt, bis das Abonnement bestätigt ist. Nach der Bestätigung des Abonnements aktualisieren Sie die Konsole, sodass der aktualisierte Status angezeigt wird.

Schritt 4: Bearbeiten eines Abrechnungsalarms

Möglicherweise möchten Sie beispielsweise den Geldbetrag, den Sie AWS pro Monat ausgeben, von 200 USD auf 400 USD erhöhen. Sie können Ihre vorhandenen Fakturierungsalarme bearbeiten und den Geldbetrag, der für das Auslösen des Alarms überschritten werden muss, erhöhen.

So bearbeiten Sie einen Fakturierungsalarm

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Ändern Sie, falls erforderlich, die Region zu USA Ost (Nord-Virginia). Die metrischen Fakturierungsdaten werden in dieser Region gespeichert und stellen die weltweiten Gebühren dar.
3. Klicken Sie im Navigationsbereich auf Alarms (Alarme).
4. Aktivieren Sie das Kontrollkästchen neben dem Alarm und wählen Sie Actions (Aktionen) und Modify (Ändern) aus.

5. Geben Sie für Wann immer meine AWS Gesamtkosten für den Monat überschritten werden, den neuen Betrag an, der überschritten werden muss, damit der Alarm ausgelöst wird, und Sie erhalten eine E-Mail-Benachrichtigung.
6. Wählen Sie Save Changes.

Schritt 5: Löschen eines Abrechnungsalarms

Wenn Sie Ihren Abrechnungsalarm nicht mehr benötigen, können Sie ihn löschen.

So löschen Sie einen Fakturierungsalarm

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Ändern Sie, falls erforderlich, die Region zu USA Ost (Nord-Virginia). Die metrischen Fakturierungsdaten werden in dieser Region gespeichert und stellen die weltweiten Gebühren dar.
3. Klicken Sie im Navigationsbereich auf Alarms (Alarme).
4. Aktivieren Sie das Kontrollkästchen neben dem Alarm und wählen Sie Actions (Aktionen) und Delete (Löschen) aus.
5. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Ja, löschen.

Szenario: Veröffentlichen Sie Metriken in CloudWatch

In diesem Szenario verwenden Sie AWS Command Line Interface (AWS CLI), um eine einzelne Metrik für eine hypothetische Anwendung mit dem Namen zu veröffentlichen. GetStarted Falls Sie den noch nicht installiert und konfiguriert haben AWS CLI, finden Sie weitere Informationen unter Getting Up with the AWS Command Line Interface im AWS Command Line Interface Benutzerhandbuch.

Aufgaben

- Schritt 1: Festlegen der Datenkonfiguration
- Schritt 2: Fügen Sie Metriken hinzu CloudWatch
- Schritt 3: Holen Sie sich Statistiken von CloudWatch
- Schritt 4: Anzeigen von Schaubildern mit der Konsole

Schritt 1: Festlegen der Datenkonfiguration

In diesem Szenario veröffentlichen Sie Datenpunkte, mit denen die Anfragelatenz für die Anwendung nachverfolgt wird. Wählen Sie einen Namen für die Metrik und den Namespace, den Sie für sinnvoll halten. Geben Sie in diesem Beispiel der Metrik einen Namen RequestLatencyund platzieren Sie alle Datenpunkte im GetStartedNamespace.

Sie veröffentlichen mehrere Datenpunkte, die zusammen 3 Stunden Latenzdaten ergeben. Die Rohdaten umfassen 15 über drei Stunden verteilte Anfragelatenz-Messwerte. Jeder Messwert wird in Millisekunden dargestellt:

- Stunde eins: 87, 51, 125, 235
- Stunde zwei: 121, 113, 189, 65, 89
- Stunde drei: 100, 47, 133, 98, 100, 328

Sie können Daten CloudWatch als einzelne Datenpunkte oder als aggregierten Satz von Datenpunkten, der als Statistiksatz bezeichnet wird, veröffentlichen. Sie können Metriken mit einer Granularität von bis zu einer Minute aggregieren. Sie können die aggregierten Datenpunkte CloudWatch als Statistiksatz mit vier vordefinierten Schlüsseln veröffentlichen:`Sum`,`Minimum`, `Maximum` und. `SampleCount`

Sie veröffentlichen die Datenpunkte aus einer Stunde als einzelne Datenpunkte. Für die Daten aus den Stunden zwei und drei aggregieren Sie die Datenpunkte und veröffentlichen eine Statistikgruppe für jede Stunde. Die wichtigsten Werte sind in der folgenden Tabelle gezeigt.

Stunde	Rohdaten	Summe	Minimum	Maximum	SampleCount
1	87				
1	51				
1	125				
1	235				
2	121, 113, 189, 65, 89	577	65	189	5

Stunde	Rohdaten	Summe	Minimum	Maximum	SampleCount
3	100, 47, 133, 98, 100, 328	806	47	328	6

Schritt 2: Fügen Sie Metriken hinzu CloudWatch

Nachdem Sie die Konfiguration Ihrer Daten festgelegt haben, können Sie die Daten hinzufügen.

Um Datenpunkte zu veröffentlichen CloudWatch

1. Führen Sie in einer Befehlszeile die folgenden put-metric-dataBefehle aus, um Daten für die erste Stunde hinzuzufügen. Ersetzen Sie den beispielhaften Zeitstempel durch einen Zeitstempel in koordinierter Weltzeit (UTC, Coordinated Universal Time), der zwei Stunden in der Vergangenheit liegt.

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 87 --unit Milliseconds
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 51 --unit Milliseconds
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 125 --unit Milliseconds
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 235 --unit Milliseconds
```

2. Fügen Sie Daten für die zweite Stunde hinzu, und verwenden Sie dazu einen Zeitstempel, der eine Stunde später als die erste Stunde liegt.

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T21:30:00Z --statistic-values
 Sum=577,Minimum=65,Maximum=189,SampleCount=5 --unit Milliseconds
```

3. Fügen Sie Daten für die dritte Stunde hinzu und lassen Sie dabei den standardmäßig auf die aktuelle Zeit eingestellten Zeitstempel weg.

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--statistic-values Sum=806,Minimum=47,Maximum=328,SampleCount=6 --unit Milliseconds
```

Schritt 3: Holen Sie sich Statistiken von CloudWatch

Nachdem Sie nun Metriken veröffentlicht haben CloudWatch, können Sie mithilfe des folgenden get-metric-statisticsBefehls Statistiken abrufen, die auf diesen Metriken basieren. Stellen Sie sicher, dass Sie die `--start-time` und `--end-time` weit genug in der Vergangenheit abgeben, um den frühesten Zeitstempel, den Sie veröffentlicht haben, mit zu erfassen.

```
aws cloudwatch get-metric-statistics --namespace GetStarted --metric-name
 RequestLatency --statistics Average \
--start-time 2016-10-14T00:00:00Z --end-time 2016-10-15T00:00:00Z --period 60
```

Das Folgende ist Ausgabebeispiel:

```
{
 "Datapoints": [],
 "Label": "Request:Latency"
}
```

Schritt 4: Anzeigen von Schaubildern mit der Konsole

Nachdem Sie Metriken veröffentlicht haben CloudWatch, können Sie die CloudWatch Konsole verwenden, um statistische Diagramme anzuzeigen.

So zeigen Sie Schaubilder Ihrer Statistiken auf der Konsole an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Bereich Navigation Metrics aus.
3. Geben Sie auf der Registerkarte Alle Metriken in das Suchfeld ein RequestLatencyund drücken Sie die Eingabetaste.
4. Aktivieren Sie das Kontrollkästchen für die RequestLatencyMetrik. Im oberen Bereich wird ein Schaubild der Metrikdaten angezeigt.

Weitere Informationen finden Sie unter Grafisches Darstellen von Metriken.

Verwendung CloudWatch mit einem SDK AWS

AWS Software Development Kits (SDKs) sind für viele gängige Programmiersprachen verfügbar. Jedes SDK bietet eine API, Codebeispiele und Dokumentation, die es Entwicklern erleichtern, Anwendungen in ihrer bevorzugten Sprache zu erstellen.

SDK-Dokumentation	Codebeispiele
AWS SDK for C++	AWS SDK for C++ Code-Beispiele
AWS CLI	AWS CLI Code-Beispiele
AWS SDK for Go	AWS SDK for Go Code-Beispiele
AWS SDK for Java	AWS SDK for Java Code-Beispiele
AWS SDK for JavaScript	AWS SDK for JavaScript Code-Beispiele
AWS SDK for Kotlin	AWS SDK for Kotlin Code-Beispiele
AWS SDK for .NET	AWS SDK for .NET Code-Beispiele
AWS SDK for PHP	AWS SDK for PHP Code-Beispiele
AWS Tools for PowerShell	Tools für PowerShell Codebeispiele
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) Code-Beispiele
AWS SDK for Ruby	AWS SDK for Ruby Code-Beispiele
AWS SDK for Rust	AWS SDK for Rust Code-Beispiele
AWS SDK für SAP ABAP	AWS SDK für SAP ABAP Code-Beispiele
AWS SDK for Swift	AWS SDK for Swift Code-Beispiele

Spezifische Beispiele für finden Sie unterCodebeispiele für die CloudWatch Verwendung von AWS SDKs. CloudWatch

Beispiel für die Verfügbarkeit

Sie können nicht finden, was Sie brauchen? Fordern Sie ein Codebeispiel an, indem Sie unten den Link Provide feedback (Feedback geben) auswählen.

Codebeispiele für die CloudWatch Verwendung von AWS SDKs

Die folgenden Codebeispiele zeigen, wie die Verwendung CloudWatch mit einem AWS Software Development Kit (SDK) funktioniert.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Während Aktionen Ihnen zeigen, wie Sie einzelne Servicefunktionen aufrufen, können Sie Aktionen im Kontext der zugehörigen Szenarien und serviceübergreifenden Beispiele sehen.

Szenarien sind Codebeispiele, die Ihnen zeigen, wie Sie eine bestimmte Aufgabe ausführen können, indem Sie mehrere Funktionen innerhalb desselben Services aufrufen.

Serviceübergreifende Beispiele sind Beispielanwendungen, die über mehrere AWS-Services hinweg arbeiten.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erste Schritte

Hallo CloudWatch

Die folgenden Codebeispiele zeigen, wie Sie mit der Verwendung beginnen CloudWatch.

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
using Amazon.CloudWatch;
using Amazon.CloudWatch.Model;
using Microsoft.Extensions.DependencyInjection;
```

```
using Microsoft.Extensions.Hosting;

namespace CloudWatchActions;

public static class HelloCloudWatch
{
    static async Task Main(string[] args)
    {
        // Use the AWS .NET Core Setup package to set up dependency injection for
 the Amazon CloudWatch service.
        // Use your AWS profile name, or leave it blank to use the default
 profile.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureServices((_, services) =>
                services.AddAWSService<IAmazonCloudWatch>()
            ).Build();

        // Now the client is available for injection.
        var cloudWatchClient =
 host.Services.GetRequiredService<IAmazonCloudWatch>();

        // You can use await and any of the async methods to get a response.
        var metricNamespace = "AWS/Billing";
        var response = await cloudWatchClient.ListMetricsAsync(new
 ListMetricsRequest
        {
            Namespace = metricNamespace
        });
        Console.WriteLine($"Hello Amazon CloudWatch! Following are some metrics
 available in the {metricNamespace} namespace:");
        Console.WriteLine();
        foreach (var metric in response.Metrics.Take(5))
        {
            Console.WriteLine($"\tMetric: {metric.MetricName}");
            Console.WriteLine($"\tNamespace: {metric.Namespace}");
            Console.WriteLine($"\tDimensions: {string.Join(", ",
 metric.Dimensions.Select(m => $"{m.Name}:{m.Value}"))}");
            Console.WriteLine();
        }
    }
}
```

- Einzelheiten zur API finden Sie ListMetricsin der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsRequest;
import software.amazon.awssdk.services.cloudwatch.paginators.ListMetricsIterable;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class HelloService {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <namespace>\s

                Where:
                  namespace - The namespace to filter against (for example, AWS/
EC2).\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        String namespace = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        listMets(cw, namespace);
        cw.close();
    }

    public static void listMets(CloudWatchClient cw, String namespace) {
        try {
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .namespace(namespace)
                    .build();

            ListMetricsIterable listRes = cw.listMetricsPaginator(request);
            listRes.stream()
                    .flatMap(r -> r.metrics().stream())
                    .forEach(metrics -> System.out.println(" Retrieved metric is:
 " + metrics.metricName()));

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie ListMetricsin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
/**
Before running this Kotlin code example, set up your development environment,
including your credentials.

For more information, see the following documentation topic:
https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html
 */
suspend fun main(args: Array<String>) {
    val usage = """
        Usage:
           <namespace>
        Where:
           namespace - The namespace to filter against (for example, AWS/EC2).
    """

    if (args.size != 1) {
        println(usage)
        exitProcess(0)
    }

    val namespace = args[0]
    listAllMets(namespace)
}

suspend fun listAllMets(namespaceVal: String?) {
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.listMetricsPaginated(request)
            .transform { it.metrics?.forEach { obj -> emit(obj) } }
            .collect { obj ->
                println("Name is ${obj.metricName}")
                println("Namespace is ${obj.namespace}")
            }
    }
}
```

- API-Details finden Sie ListMetricsin der API-Referenz zum AWS SDK für Kotlin.

Codebeispiele

- Aktionen zur CloudWatch Verwendung von SDKs AWS
  - Verwendung DeleteAlarms mit einem AWS SDK oder CLI
  - Verwendung DeleteAnomalyDetector mit einem AWS SDK oder CLI
  - Verwendung DeleteDashboards mit einem AWS SDK oder CLI
  - Verwendung DescribeAlarmHistory mit einem AWS SDK oder CLI
  - Verwendung DescribeAlarms mit einem AWS SDK oder CLI
  - Verwendung DescribeAlarmsForMetric mit einem AWS SDK oder CLI
  - Verwendung DescribeAnomalyDetectors mit einem AWS SDK oder CLI
  - Verwendung DisableAlarmActions mit einem AWS SDK oder CLI
  - Verwendung EnableAlarmActions mit einem AWS SDK oder CLI
  - Verwendung GetDashboard mit einem AWS SDK oder CLI
  - Verwendung GetMetricData mit einem AWS SDK oder CLI
  - Verwendung GetMetricStatistics mit einem AWS SDK oder CLI
  - Verwendung GetMetricWidgetImage mit einem AWS SDK oder CLI
  - Verwendung ListDashboards mit einem AWS SDK oder CLI
  - Verwendung ListMetrics mit einem AWS SDK oder CLI
  - Verwendung PutAnomalyDetector mit einem AWS SDK oder CLI
  - Verwendung PutDashboard mit einem AWS SDK oder CLI
  - Verwendung PutMetricAlarm mit einem AWS SDK oder CLI
  - Verwendung PutMetricData mit einem AWS SDK oder CLI
- Szenarien für die CloudWatch Verwendung von AWS SDKs
  - Erste Schritte mit CloudWatch Alarmen mithilfe eines AWS SDK
  - Erste Schritte mit CloudWatch Metriken, Dashboards und Alarmen mithilfe eines SDK AWS
  - CloudWatch Metriken und Alarme mithilfe eines AWS SDK verwalten
- Serviceübergreifende Beispiele für die CloudWatch Verwendung von SDKs AWS
  - Überwachen Sie die Leistung von Amazon DynamoDB mithilfe eines SDK AWS

Aktionen zur CloudWatch Verwendung von SDKs AWS

Die folgenden Codebeispiele zeigen, wie einzelne CloudWatch Aktionen mit AWS SDKs ausgeführt werden. Diese Auszüge rufen die CloudWatch API auf und sind Codeauszüge aus größeren Programmen, die im Kontext ausgeführt werden müssen. Jedes Beispiel enthält einen Link zu GitHub, wo Sie Anweisungen zum Einrichten und Ausführen des Codes finden.

Die folgenden Beispiele enthalten nur die am häufigsten verwendeten Aktionen. Eine vollständige Liste finden Sie in der Amazon CloudWatch API-Referenz.

Beispiele

- Verwendung DeleteAlarms mit einem AWS SDK oder CLI
- Verwendung DeleteAnomalyDetector mit einem AWS SDK oder CLI
- Verwendung DeleteDashboards mit einem AWS SDK oder CLI
- Verwendung DescribeAlarmHistory mit einem AWS SDK oder CLI
- Verwendung DescribeAlarms mit einem AWS SDK oder CLI
- Verwendung DescribeAlarmsForMetric mit einem AWS SDK oder CLI
- Verwendung DescribeAnomalyDetectors mit einem AWS SDK oder CLI
- Verwendung DisableAlarmActions mit einem AWS SDK oder CLI
- Verwendung EnableAlarmActions mit einem AWS SDK oder CLI
- Verwendung GetDashboard mit einem AWS SDK oder CLI
- Verwendung GetMetricData mit einem AWS SDK oder CLI
- Verwendung GetMetricStatistics mit einem AWS SDK oder CLI
- Verwendung GetMetricWidgetImage mit einem AWS SDK oder CLI
- Verwendung ListDashboards mit einem AWS SDK oder CLI
- Verwendung ListMetrics mit einem AWS SDK oder CLI
- Verwendung PutAnomalyDetector mit einem AWS SDK oder CLI
- Verwendung PutDashboard mit einem AWS SDK oder CLI
- Verwendung PutMetricAlarm mit einem AWS SDK oder CLI
- Verwendung PutMetricData mit einem AWS SDK oder CLI

Verwendung **`DeleteAlarms`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DeleteAlarms`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- Erste Schritte mit Alarmen
- Erste Schritte mit CloudWatch-Metriken, -Dashboards und -Alarmen
- Metriken und Alarme verwalten

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
/// <summary>
/// Delete a list of alarms from CloudWatch.
/// </summary>
/// <param name="alarmNames">A list of names of alarms to delete.</param>
/// <returns>True if successful.</returns>
public async Task<bool> DeleteAlarms(List<string> alarmNames)
{
    var deleteAlarmsResult = await _amazonCloudWatch.DeleteAlarmsAsync(
        new DeleteAlarmsRequest()
        {
            AlarmNames = alarmNames
        });

    return deleteAlarmsResult.HttpStatusCode == HttpStatusCode.OK;
}
```

- Einzelheiten zur API finden Sie DeleteAlarmsin der AWS SDK for .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Binden Sie die erforderlichen Dateien ein.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/DeleteAlarmsRequest.h>
#include <iostream>
```

Löschen Sie den Alarm.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::DeleteAlarmsRequest request;
        request.AddAlarmNames(alarm_name);

        auto outcome = cw.DeleteAlarms(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to delete CloudWatch alarm:" <<
                outcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout << "Successfully deleted CloudWatch alarm " << alarm_name
                << std::endl;
        }
```

- Einzelheiten zur API finden Sie DeleteAlarmsin der AWS SDK for C++ API-Referenz.

CLI

AWS CLI

So löschen Sie einen Alarm

Im folgenden Beispiel wird der `delete-alarms` Befehl verwendet, um den CloudWatch Amazon-Alarm mit dem Namen „myalarm“ zu löschen:

```
aws cloudwatch delete-alarms --alarm-names myalarm
```

Ausgabe:

```
This command returns to the prompt if successful.
```

- Einzelheiten zur API finden Sie DeleteAlarmsin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.DeleteAlarmsRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
```

```
public class DeleteAlarm {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <alarmName>

                Where:
                  alarmName - An alarm name to delete (for example, MyAlarm).
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alarmName = args[0];
        Region region = Region.US_EAST_2;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        deleteCWAlarm(cw, alarmName);
        cw.close();
    }

    public static void deleteCWAlarm(CloudWatchClient cw, String alarmName) {
        try {
            DeleteAlarmsRequest request = DeleteAlarmsRequest.builder()
                    .alarmNames(alarmName)
                    .build();

            cw.deleteAlarms(request);
            System.out.printf("Successfully deleted alarm %s", alarmName);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie DeleteAlarmsin der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Importieren Sie das SDK- und Client-Module und rufen Sie die API auf.

```
import { DeleteAlarmsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DeleteAlarmsCommand({
    AlarmNames: [process.env.CLOUDWATCH_ALARM_NAME], // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Erstellen Sie den Client in einem separaten Modul und exportieren Sie ihn.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Weitere Informationen finden Sie im AWS SDK for JavaScript -Entwicklerhandbuch.

- Einzelheiten zur API finden Sie DeleteAlarmsin der AWS SDK for JavaScript API-Referenz.

SDK für JavaScript (v2)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Importieren Sie das SDK- und Client-Module und rufen Sie die API auf.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  AlarmNames: ["Web_Server_CPU_Utilization"],
};

cw.deleteAlarms(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Weitere Informationen finden Sie im AWS SDK for JavaScript -Entwicklerhandbuch.
- Einzelheiten zur API finden Sie DeleteAlarmsin der AWS SDK for JavaScript API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun deleteAlarm(alarmNameVal: String) {
    val request = DeleteAlarmsRequest {
        alarmNames = listOf(alarmNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAlarms(request)
        println("Successfully deleted alarm $alarmNameVal")
    }
}
```

- API-Details finden Sie DeleteAlarmsin der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
```

```
        """
        self.cloudwatch_resource = cloudwatch_resource


    def delete_metric_alarms(self, metric_namespace, metric_name):
        """
        Deletes all of the alarms that are currently watching the specified
metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(metric_namespace,
metric_name)
            metric.alarms.delete()
            logger.info(
                "Deleted alarms for metric %s.%s.", metric_namespace, metric_name
            )
        except ClientError:
            logger.exception(
                "Couldn't delete alarms for metric %s.%s.",
                metric_namespace,
                metric_name,
            )
            raise
```

- Einzelheiten zur API finden Sie DeleteAlarmsin AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    TRY.
        lo_cwt->deletealarms(
          it_alarmnames = it_alarm_names
        ).
        MESSAGE 'Alarms deleted.' TYPE 'I'.
      CATCH /aws1/cx_cwtresourcenotfound .
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    ENDTRY.
```

- Einzelheiten zur API finden Sie DeleteAlarmsin der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DeleteAnomalyDetector`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DeleteAnomalyDetector`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erste Schritte mit CloudWatch-Metriken, -Dashboards und -Alarmen

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    /// <summary>
    /// Delete a single metric anomaly detector.
    /// </summary>
```

```
    /// <param name="anomalyDetector">The anomaly detector to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteAnomalyDetector(SingleMetricAnomalyDetector
anomalyDetector)
    {
        var deleteAnomalyDetectorResponse = await
_amazonCloudWatch.DeleteAnomalyDetectorAsync(
            new DeleteAnomalyDetectorRequest()
            {
                SingleMetricAnomalyDetector = anomalyDetector
            });

        return deleteAnomalyDetectorResponse.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Einzelheiten zur API finden Sie DeleteAnomalyDetectorin der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    public static void deleteAnomalyDetector(CloudWatchClient cw, String
fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();
```

```
            SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .stat("Maximum")
                    .build();

            DeleteAnomalyDetectorRequest request =
DeleteAnomalyDetectorRequest.builder()
                    .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                    .build();

            cw.deleteAnomalyDetector(request);
            System.out.println("Successfully deleted the Anomaly Detector.");

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
```

- Einzelheiten zur API finden Sie DeleteAnomalyDetectorin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun deleteAnomalyDetector(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
```

```
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val request = DeleteAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAnomalyDetector(request)
        println("Successfully deleted the Anomaly Detector.")
    }
}
```

- API-Details finden Sie DeleteAnomalyDetectorin der API-Referenz zum AWS SDK für Kotlin.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DeleteDashboards`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DeleteDashboards`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erste Schritte mit CloudWatch-Metriken, -Dashboards und -Alarmen

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    /// <summary>
    /// Delete a list of CloudWatch dashboards.
    /// </summary>
    /// <param name="dashboardNames">List of dashboard names to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteDashboards(List<string> dashboardNames)
    {
        var deleteDashboardsResponse = await
_amazonCloudWatch.DeleteDashboardsAsync(
            new DeleteDashboardsRequest()
            {
                DashboardNames = dashboardNames
            });

        return deleteDashboardsResponse.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Einzelheiten zur API finden Sie DeleteDashboardsin der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    public static void deleteDashboard(CloudWatchClient cw, String dashboardName)
{
        try {
            DeleteDashboardsRequest dashboardsRequest =
DeleteDashboardsRequest.builder()
                    .dashboardNames(dashboardName)
                    .build();
            cw.deleteDashboards(dashboardsRequest);
            System.out.println(dashboardName + " was successfully deleted.");

        } catch (CloudWatchException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie DeleteDashboardsin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun deleteDashboard(dashboardName: String) {
    val dashboardsRequest = DeleteDashboardsRequest {
        dashboardNames = listOf(dashboardName)
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteDashboards(dashboardsRequest)
        println("$dashboardName was successfully deleted.")
    }
}
```

- API-Details finden Sie DeleteDashboardsin der API-Referenz zum AWS SDK für Kotlin.

PowerShell

Tools für PowerShell

Beispiel 1: Löscht das angegebene Dashboard und lädt zur Bestätigung ein, bevor Sie fortfahren. Um die Bestätigung zu umgehen, fügen Sie dem Befehl den Schalter -Force hinzu.

```
Remove-CWDashboard -DashboardName Dashboard1
```

- Einzelheiten zur API finden Sie unter DeleteDashboards AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwendung CloudWatch mit einem SDK AWS Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeAlarmHistory`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeAlarmHistory`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erste Schritte mit CloudWatch-Metriken, -Dashboards und -Alarmen

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    /// <summary>
    /// Describe the history of an alarm for a number of days in the past.
```

```
    /// </summary>
    /// <param name="alarmName">The name of the alarm.</param>
    /// <param name="historyDays">The number of days in the past.</param>
    /// <returns>The list of alarm history data.</returns>
    public async Task<List<AlarmHistoryItem>> DescribeAlarmHistory(string
alarmName, int historyDays)
    {
        List<AlarmHistoryItem> alarmHistory = new List<AlarmHistoryItem>();
        var paginatedAlarmHistory =
_amazonCloudWatch.Paginators.DescribeAlarmHistory(
            new DescribeAlarmHistoryRequest()
            {
                AlarmName = alarmName,
                EndDateUtc = DateTime.UtcNow,
                HistoryItemType = HistoryItemType.StateUpdate,
                StartDateUtc = DateTime.UtcNow.AddDays(-historyDays)
            });

        await foreach (var data in paginatedAlarmHistory.AlarmHistoryItems)
        {
            alarmHistory.Add(data);
        }
        return alarmHistory;
    }
```

- Einzelheiten zur API finden Sie DescribeAlarmHistoryin der AWS SDK for .NET API-Referenz.

CLI

AWS CLI

So rufen Sie den Verlauf eines Alarms ab

Im folgenden Beispiel wird der `describe-alarm-history` Befehl verwendet, um den Verlauf für den CloudWatch Amazon-Alarm mit dem Namen „myalarm“ abzurufen:

```
aws cloudwatch describe-alarm-history --alarm-name "myalarm" --history-item-type
 StateUpdate
```

Ausgabe:

```
{
    "AlarmHistoryItems": [
        {
            "Timestamp": "2014-04-09T18:59:06.442Z",
            "HistoryItemType": "StateUpdate",
            "AlarmName": "myalarm",
            "HistoryData": "{\"version\":\"1.0\",\"oldState\":{\"stateValue
\":\"ALARM\",\"stateReason\":\"testing purposes\"},\"newState\":{\"stateValue
\":\"OK\",\"stateReason\":\"Threshold Crossed: 2 datapoints were not
 greater than the threshold (70.0). The most recent datapoints: [38.958,
 40.292].\",\"stateReasonData\":{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}}}",
            "HistorySummary": "Alarm updated from ALARM to OK"
        },
        {
            "Timestamp": "2014-04-09T18:59:05.805Z",
            "HistoryItemType": "StateUpdate",
            "AlarmName": "myalarm",
            "HistoryData": "{\"version\":\"1.0\",\"oldState\":{\"stateValue
\":\"OK\",\"stateReason\":\"Threshold Crossed: 2 datapoints were
 not greater than the threshold (70.0). The most recent datapoints:
 [38.839999999999996, 39.714].\",\"stateReasonData\":{\"version\":
\"1.0\",\"queryDate\":\"2014-03-11T22:45:41.569+0000\",\"startDate\":
\"2014-03-11T22:30:00.000+0000\",\"statistic\":\"Average\",\"period\":300,
\"recentDatapoints\":[38.839999999999996,39.714],\"threshold\":70.0}},\"newState
\":{\"stateValue\":\"ALARM\",\"stateReason\":\"testing purposes\"}}",
            "HistorySummary": "Alarm updated from OK to ALARM"
        }
    ]
}
```

- Einzelheiten zur API finden Sie DescribeAlarmHistoryin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
   public static void getAlarmHistory(CloudWatchClient cw, String fileName,
String date) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String alarmName = rootNode.findValue("exampleAlarmName").asText();

           Instant start = Instant.parse(date);
           Instant endDate = Instant.now();
           DescribeAlarmHistoryRequest historyRequest =
DescribeAlarmHistoryRequest.builder()
                   .startDate(start)
                   .endDate(endDate)
                   .alarmName(alarmName)
                   .historyItemType(HistoryItemType.ACTION)
                   .build();

           DescribeAlarmHistoryResponse response =
cw.describeAlarmHistory(historyRequest);
           List<AlarmHistoryItem> historyItems = response.alarmHistoryItems();
           if (historyItems.isEmpty()) {
               System.out.println("No alarm history data found for " + alarmName
+ ".");
           } else {
               for (AlarmHistoryItem item : historyItems) {
                   System.out.println("History summary: " +
item.historySummary());
                   System.out.println("Time stamp: " + item.timestamp());
               }
```

```
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie DescribeAlarmHistoryin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun getAlarmHistory(fileName: String, date: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val alarmNameVal = rootNode.findValue("exampleAlarmName").asText()
    val start = Instant.parse(date)
    val endDateVal = Instant.now()

    val historyRequest = DescribeAlarmHistoryRequest {
        startDate = aws.smithy.kotlin.runtime.time.Instant(start)
        endDate = aws.smithy.kotlin.runtime.time.Instant(endDateVal)
        alarmName = alarmNameVal
        historyItemType = HistoryItemType.Action
    }

    CloudWatchClient { credentialsProvider = EnvironmentCredentialsProvider();
 region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarmHistory(historyRequest)
        val historyItems = response.alarmHistoryItems
```

```
        if (historyItems != null) {
            if (historyItems.isEmpty()) {
                println("No alarm history data found for $alarmNameVal.")
            } else {
                for (item in historyItems) {
                    println("History summary ${item.historySummary}")
                    println("Time stamp: ${item.timestamp}")
                }
            }
        }
    }
}
```

- API-Details finden Sie DescribeAlarmHistoryin der API-Referenz zum AWS SDK für Kotlin.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeAlarms`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeAlarms`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- Erste Schritte mit Alarmen
- Erste Schritte mit CloudWatch-Metriken, -Dashboards und -Alarmen

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    /// <summary>
    /// Describe the current alarms, optionally filtered by state.
    /// </summary>
    /// <param name="stateValue">Optional filter for alarm state.</param>
    /// <returns>The list of alarm data.</returns>
    public async Task<List<MetricAlarm>> DescribeAlarms(StateValue? stateValue =
null)
    {
        List<MetricAlarm> alarms = new List<MetricAlarm>();
        var paginatedDescribeAlarms =
_amazonCloudWatch.Paginators.DescribeAlarms(
            new DescribeAlarmsRequest()
            {
                StateValue = stateValue
            });

        await foreach (var data in paginatedDescribeAlarms.MetricAlarms)
        {
            alarms.Add(data);
        }
        return alarms;
    }
```

- Einzelheiten zur API finden Sie DescribeAlarmsin der AWS SDK for .NET API-Referenz.

CLI

AWS CLI

So listen Sie Informationen über einen Alarm auf

Im folgenden Beispiel wird der `describe-alarms`-Befehl verwendet, um Informationen über den Alarm mit dem Namen „myalarm“ bereitzustellen:

```
aws cloudwatch describe-alarms --alarm-names "myalarm"
```

Ausgabe:

```
{
    "MetricAlarms": [
```

```
        {
            "EvaluationPeriods": 2,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:123456789012:alarm:myalarm",
            "StateUpdatedTimestamp": "2014-04-09T18:59:06.442Z",
            "AlarmConfigurationUpdatedTimestamp": "2012-12-27T00:49:54.032Z",
            "ComparisonOperator": "GreaterThanThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:123456789012:myHighCpuAlarm"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myalarm",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 2 datapoints were not greater than
 the threshold (70.0). The most recent datapoints: [38.958, 40.292].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": true,
            "MetricName": "CPUUtilization"
        }
    ]
}
```

- Einzelheiten zur API finden Sie DescribeAlarmsin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    public static void describeAlarms(CloudWatchClient cw) {
        try {
            List<AlarmType> typeList = new ArrayList<>();
            typeList.add(AlarmType.METRIC_ALARM);

            DescribeAlarmsRequest alarmsRequest = DescribeAlarmsRequest.builder()
                    .alarmTypes(typeList)
                    .maxRecords(10)
                    .build();

            DescribeAlarmsResponse response = cw.describeAlarms(alarmsRequest);
            List<MetricAlarm> alarmList = response.metricAlarms();
            for (MetricAlarm alarm : alarmList) {
                System.out.println("Alarm name: " + alarm.alarmName());
                System.out.println("Alarm description: " +
alarm.alarmDescription());
            }
        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie DescribeAlarmsin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun describeAlarms() {
    val typeList = ArrayList<AlarmType>()
    typeList.add(AlarmType.MetricAlarm)
    val alarmsRequest = DescribeAlarmsRequest {
        alarmTypes = typeList
        maxRecords = 10
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarms(alarmsRequest)
        response.metricAlarms?.forEach { alarm ->
            println("Alarm name: ${alarm.alarmName}")
            println("Alarm description: ${alarm.alarmDescription}")
        }
    }
}
```

- API-Details finden Sie DescribeAlarmsin der API-Referenz zum AWS SDK für Kotlin.

Ruby

SDK für Ruby

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
require "aws-sdk-cloudwatch"

# Lists the names of available Amazon CloudWatch alarms.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @example
#   list_alarms(Aws::CloudWatch::Client.new(region: 'us-east-1'))
def list_alarms(cloudwatch_client)
  response = cloudwatch_client.describe_alarms
  if response.metric_alarms.count.positive?
    response.metric_alarms.each do |alarm|
      puts alarm.alarm_name
    end
  else
    puts "No alarms found."
  end
rescue StandardError => e
  puts "Error getting information about alarms: #{e.message}"
end
```

- Einzelheiten zur API finden Sie DescribeAlarmsin der AWS SDK for Ruby API-Referenz.

SAP ABAP

SDK für SAP ABAP

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    TRY.
        oo_result = lo_cwt->describealarms(                " oo_result is
 returned for testing purposes. "
          it_alarmnames = it_alarm_names
        ).
        MESSAGE 'Alarms retrieved.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
```

```
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Einzelheiten zur API finden Sie DescribeAlarmsin der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeAlarmsForMetric`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeAlarmsForMetric`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- Erste Schritte mit CloudWatch-Metriken, -Dashboards und -Alarmen
- Metriken und Alarme verwalten

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    /// <summary>
    /// Describe the current alarms for a specific metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The list of alarm data.</returns>
```

```
    public async Task<List<MetricAlarm>> DescribeAlarmsForMetric(string
metricNamespace, string metricName)
    {
        var alarmsResult = await _amazonCloudWatch.DescribeAlarmsForMetricAsync(
            new DescribeAlarmsForMetricRequest()
            {
                Namespace = metricNamespace,
                MetricName = metricName
            });

        return alarmsResult.MetricAlarms;
    }
```

- Einzelheiten zur API finden Sie DescribeAlarmsForMetricin der AWS SDK for .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Binden Sie die erforderlichen Dateien ein.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/DescribeAlarmsRequest.h>
#include <aws/monitoring/model/DescribeAlarmsResult.h>
#include <iomanip>
#include <iostream>
```

Beschreiben Sie die Alarme.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::DescribeAlarmsRequest request;
```

```
        request.SetMaxRecords(1);

        bool done = false;
        bool header = false;
        while (!done)
        {
            auto outcome = cw.DescribeAlarms(request);
            if (!outcome.IsSuccess())
            {
                std::cout << "Failed to describe CloudWatch alarms:" <<
                    outcome.GetError().GetMessage() << std::endl;
                break;
            }

            if (!header)
            {
                std::cout << std::left <<
                    std::setw(32) << "Name" <<
                    std::setw(64) << "Arn" <<
                    std::setw(64) << "Description" <<
                    std::setw(20) << "LastUpdated" <<
                    std::endl;
                header = true;
            }

            const auto &alarms = outcome.GetResult().GetMetricAlarms();
            for (const auto &alarm : alarms)
            {
                std::cout << std::left <<
                    std::setw(32) << alarm.GetAlarmName() <<
                    std::setw(64) << alarm.GetAlarmArn() <<
                    std::setw(64) << alarm.GetAlarmDescription() <<
                    std::setw(20) <<
                    alarm.GetAlarmConfigurationUpdatedTimestamp().ToGmtString(
                        SIMPLE_DATE_FORMAT_STR) <<
                    std::endl;
            }

            const auto &next_token = outcome.GetResult().GetNextToken();
            request.SetNextToken(next_token);
            done = next_token.empty();
        }
```

- Einzelheiten zur API finden Sie DescribeAlarmsForMetricin der AWS SDK for C++ API-Referenz.

CLI

AWS CLI

So zeigen Sie Informationen über Alarme an, die einer Metrik zugeordnet sind

Im folgenden Beispiel wird der `describe-alarms-for-metric`-Befehl verwendet, um Informationen über alle Alarme anzuzeigen, die der Amazon-EC2-Metrik CPUUtilization und der Instance mit der ID i-0c986c72 zugeordnet sind:

```
aws cloudwatch describe-alarms-for-metric --metric-name CPUUtilization --
namespace AWS/EC2 --dimensions Name=InstanceId,Value=i-0c986c72
```

Ausgabe:

```
{
    "MetricAlarms": [
        {
            "EvaluationPeriods": 10,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:111122223333:alarm:myHighCpuAlarm2",
            "StateUpdatedTimestamp": "2013-10-30T03:03:51.479Z",
            "AlarmConfigurationUpdatedTimestamp": "2013-10-30T03:03:50.865Z",
            "ComparisonOperator": "GreaterThanOrEqualToThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:111122223333:NotifyMe"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2013-10-30T03:03:51.479+0000\",\"startDate\":\"2013-10-30T02:08:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":
[40.698,39.612,42.432,39.796,38.816,42.28,42.854,40.088,40.760000000000005,41.316],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myHighCpuAlarm2",
            "Dimensions": [
```

```
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 10 datapoints were not
 greater than or equal to the threshold (70.0). The most recent datapoints:
 [40.760000000000005, 41.316].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": true,
            "MetricName": "CPUUtilization"
        },
        {
            "EvaluationPeriods": 2,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:111122223333:alarm:myHighCpuAlarm",
            "StateUpdatedTimestamp": "2014-04-09T18:59:06.442Z",
            "AlarmConfigurationUpdatedTimestamp": "2014-04-09T22:26:05.958Z",
            "ComparisonOperator": "GreaterThanThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:111122223333:HighCPUAlarm"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myHighCpuAlarm",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 2 datapoints were not greater than
 the threshold (70.0). The most recent datapoints: [38.958, 40.292].",
            "InsufficientDataActions": [],
```

```
            "OKActions": [],
            "ActionsEnabled": false,
            "MetricName": "CPUUtilization"
        }
    ]
}
```

- Einzelheiten zur API finden Sie DescribeAlarmsForMetricin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
   public static void checkForMetricAlarm(CloudWatchClient cw, String fileName)
{
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();
           boolean hasAlarm = false;
           int retries = 10;

           DescribeAlarmsForMetricRequest metricRequest =
DescribeAlarmsForMetricRequest.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .build();

           while (!hasAlarm && retries > 0) {
```

```
                DescribeAlarmsForMetricResponse response =
cw.describeAlarmsForMetric(metricRequest);
                hasAlarm = response.hasMetricAlarms();
                retries--;
                Thread.sleep(20000);
                System.out.println(".");
            }
            if (!hasAlarm)
                System.out.println("No Alarm state found for " + customMetricName
+ " after 10 retries.");
            else
                System.out.println("Alarm state found for " + customMetricName +
".");

        } catch (CloudWatchException | IOException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie DescribeAlarmsForMetricin der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Importieren Sie das SDK- und Client-Module und rufen Sie die API auf.

```
import { DescribeAlarmsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DescribeAlarmsCommand({
```

```
    AlarmNames: [process.env.CLOUDWATCH_ALARM_NAME], // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Erstellen Sie den Client in einem separaten Modul und exportieren Sie ihn.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Weitere Informationen finden Sie im AWS SDK for JavaScript -Entwicklerhandbuch.
- Einzelheiten zur API finden Sie DescribeAlarmsForMetricin der AWS SDK for JavaScript API-Referenz.

SDK für JavaScript (v2)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

cw.describeAlarms({ StateValue: "INSUFFICIENT_DATA" }, function (err, data) {
```

```
  if (err) {
    console.log("Error", err);
  } else {
    // List the names of all current alarms in the console
    data.MetricAlarms.forEach(function (item, index, array) {
      console.log(item.AlarmName);
    });
  }
});
```

- Weitere Informationen finden Sie im AWS SDK for JavaScript -Entwicklerhandbuch.
- Einzelheiten zur API finden Sie DescribeAlarmsForMetricin der AWS SDK for JavaScript API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun checkForMetricAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()
    var hasAlarm = false
    var retries = 10

    val metricRequest = DescribeAlarmsForMetricRequest {
        metricName = customMetricName
        namespace = customMetricNamespace
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        while (!hasAlarm && retries > 0) {
```

```
            val response = cwClient.describeAlarmsForMetric(metricRequest)
            if (response.metricAlarms?.count()!! > 0) {
                hasAlarm = true
            }
            retries--
            delay(20000)
            println(".")
        }
        if (!hasAlarm) println("No Alarm state found for $customMetricName after
 10 retries.") else println("Alarm state found for $customMetricName.")
    }
}
```

- API-Details finden Sie DescribeAlarmsForMetricin der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def get_metric_alarms(self, metric_namespace, metric_name):
        """
        Gets the alarms that are currently watching the specified metric.
```

```
        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        :returns: An iterator that yields the alarms.
        """
        metric = self.cloudwatch_resource.Metric(metric_namespace, metric_name)
        alarm_iter = metric.alarms.all()
        logger.info("Got alarms for metric %s.%s.", metric_namespace,
metric_name)
        return alarm_iter
```

- Einzelheiten zur API finden Sie DescribeAlarmsForMetricin AWS SDK for Python (Boto3) API Reference.

Ruby

SDK für Ruby

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @example
#   describe_metric_alarms(Aws::CloudWatch::Client.new(region: 'us-east-1'))
def describe_metric_alarms(cloudwatch_client)
  response = cloudwatch_client.describe_alarms

  if response.metric_alarms.count.positive?
    response.metric_alarms.each do |alarm|
      puts "-" * 16
      puts "Name:           " + alarm.alarm_name
      puts "State value:    " + alarm.state_value
      puts "State reason:   " + alarm.state_reason
      puts "Metric:         " + alarm.metric_name
```

```
      puts "Namespace:      " + alarm.namespace
      puts "Statistic:      " + alarm.statistic
      puts "Period:         " + alarm.period.to_s
      puts "Unit:           " + alarm.unit.to_s
      puts "Eval. periods:  " + alarm.evaluation_periods.to_s
      puts "Threshold:      " + alarm.threshold.to_s
      puts "Comp. operator: " + alarm.comparison_operator

      if alarm.key?(:ok_actions) && alarm.ok_actions.count.positive?
        puts "OK actions:"
        alarm.ok_actions.each do |a|
          puts "  " + a
        end
      end

      if alarm.key?(:alarm_actions) && alarm.alarm_actions.count.positive?
        puts "Alarm actions:"
        alarm.alarm_actions.each do |a|
          puts "  " + a
        end
      end

      if alarm.key?(:insufficient_data_actions) &&
          alarm.insufficient_data_actions.count.positive?
        puts "Insufficient data actions:"
        alarm.insufficient_data_actions.each do |a|
          puts "  " + a
        end
      end

      puts "Dimensions:"
      if alarm.key?(:dimensions) && alarm.dimensions.count.positive?
        alarm.dimensions.each do |d|
          puts "  Name: " + d.name + ", Value: " + d.value
        end
      else
        puts "  None for this alarm."
      end
    end
  else
    puts "No alarms found."
  end
rescue StandardError => e
  puts "Error getting information about alarms: #{e.message}"
```

```
end

# Example usage:
def run_me
  region = ""

  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage:   ruby cw-ruby-example-show-alarms.rb REGION"
    puts "Example: ruby cw-ruby-example-show-alarms.rb us-east-1"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    region = "us-east-1"
  # Otherwise, use the values as specified at the command prompt.
  else
    region = ARGV[0]
  end

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)
  puts "Available alarms:"
  describe_metric_alarms(cloudwatch_client)
end

run_me if $PROGRAM_NAME == __FILE__
```

- Einzelheiten zur API finden Sie DescribeAlarmsForMetricin der AWS SDK for Ruby API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeAnomalyDetectors`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DescribeAnomalyDetectors`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erste Schritte mit CloudWatch-Metriken, -Dashboards und -Alarmen

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    /// <summary>
    /// Describe anomaly detectors for a metric and namespace.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The metric of the anomaly detectors.</param>
    /// <returns>The list of detectors.</returns>
    public async Task<List<AnomalyDetector>> DescribeAnomalyDetectors(string
metricNamespace, string metricName)
    {
        List<AnomalyDetector> detectors = new List<AnomalyDetector>();
        var paginatedDescribeAnomalyDetectors =
_amazonCloudWatch.Paginators.DescribeAnomalyDetectors(
            new DescribeAnomalyDetectorsRequest()
            {
                MetricName = metricName,
                Namespace = metricNamespace
            });

        await foreach (var data in
paginatedDescribeAnomalyDetectors.AnomalyDetectors)
        {
            detectors.Add(data);
        }

        return detectors;
    }
```

- Einzelheiten zur API finden Sie DescribeAnomalyDetectorsin der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
   public static void describeAnomalyDetectors(CloudWatchClient cw, String
fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();
           DescribeAnomalyDetectorsRequest detectorsRequest =
DescribeAnomalyDetectorsRequest.builder()
                   .maxResults(10)
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .build();

           DescribeAnomalyDetectorsResponse response =
cw.describeAnomalyDetectors(detectorsRequest);
           List<AnomalyDetector> anomalyDetectorList =
response.anomalyDetectors();
           for (AnomalyDetector detector : anomalyDetectorList) {
               System.out.println("Metric name: " +
detector.singleMetricAnomalyDetector().metricName());
```

```
                System.out.println("State: " + detector.stateValue());
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie DescribeAnomalyDetectorsin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun describeAnomalyDetectors(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val detectorsRequest = DescribeAnomalyDetectorsRequest {
        maxResults = 10
        metricName = customMetricName
        namespace = customMetricNamespace
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAnomalyDetectors(detectorsRequest)
        response.anomalyDetectors?.forEach { detector ->
            println("Metric name:
 ${detector.singleMetricAnomalyDetector?.metricName}")
```

```
            println("State: ${detector.stateValue}")
        }
    }
}
```

- API-Details finden Sie DescribeAnomalyDetectorsin der API-Referenz zum AWS SDK für Kotlin.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DisableAlarmActions`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`DisableAlarmActions`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- Erste Schritte mit Alarmen
- Metriken und Alarme verwalten

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    /// <summary>
    /// Disable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DisableAlarmActions(List<string> alarmNames)
```

```
    {
        var disableAlarmActionsResult = await
_amazonCloudWatch.DisableAlarmActionsAsync(
            new DisableAlarmActionsRequest()
            {
                AlarmNames = alarmNames
            });

        return disableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Einzelheiten zur API finden Sie DisableAlarmActionsin der AWS SDK for .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Binden Sie die erforderlichen Dateien ein.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/DisableAlarmActionsRequest.h>
#include <iostream>
```

Deaktivieren der Alarmaktionen

```
        Aws::CloudWatch::CloudWatchClient cw;

        Aws::CloudWatch::Model::DisableAlarmActionsRequest
 disableAlarmActionsRequest;
        disableAlarmActionsRequest.AddAlarmNames(alarm_name);
```

```
        auto disableAlarmActionsOutcome =
cw.DisableAlarmActions(disableAlarmActionsRequest);
        if (!disableAlarmActionsOutcome.IsSuccess())
        {
            std::cout << "Failed to disable actions for alarm " << alarm_name <<
                ": " << disableAlarmActionsOutcome.GetError().GetMessage() <<
                std::endl;
        }
        else
        {
            std::cout << "Successfully disabled actions for alarm " <<
                alarm_name << std::endl;
        }
```

- Einzelheiten zur API finden Sie DisableAlarmActionsin der AWS SDK for C++ API-Referenz.

CLI

AWS CLI

So deaktivieren Sie Aktionen für einen Alarm

Das folgende Beispiel verwendet den `disable-alarm-actions`-Befehl, um alle Aktionen für den Alarm mit dem Namen „myalarm“ zu deaktivieren:

```
aws cloudwatch disable-alarm-actions --alarm-names myalarm
```

Wenn dieser Befehl erfolgreich war, kehrt er zur Eingabeaufforderung zurück.

- Einzelheiten zur API finden Sie DisableAlarmActionsin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import
 software.amazon.awssdk.services.cloudwatch.model.DisableAlarmActionsRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DisableAlarmActions {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <alarmName>

                Where:
                  alarmName - An alarm name to disable (for example, MyAlarm).
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alarmName = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        disableActions(cw, alarmName);
        cw.close();
    }

    public static void disableActions(CloudWatchClient cw, String alarmName) {
        try {
```

```
            DisableAlarmActionsRequest request =
 DisableAlarmActionsRequest.builder()
                    .alarmNames(alarmName)
                    .build();

            cw.disableAlarmActions(request);
            System.out.printf("Successfully disabled actions on alarm %s",
 alarmName);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie DisableAlarmActionsin der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Importieren Sie das SDK- und Client-Module und rufen Sie die API auf.

```
import { DisableAlarmActionsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DisableAlarmActionsCommand({
    AlarmNames: process.env.CLOUDWATCH_ALARM_NAME, // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });
```

```
  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Erstellen Sie den Client in einem separaten Modul und exportieren Sie ihn.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Weitere Informationen finden Sie im AWS SDK for JavaScript -Entwicklerhandbuch.
- Einzelheiten zur API finden Sie DisableAlarmActionsin der AWS SDK for JavaScript API-Referenz.

SDK für JavaScript (v2)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Importieren Sie das SDK- und Client-Module und rufen Sie die API auf.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

cw.disableAlarmActions(
  { AlarmNames: ["Web_Server_CPU_Utilization"] },
```

```
  function (err, data) {
    if (err) {
      console.log("Error", err);
    } else {
      console.log("Success", data);
    }
  }
);
```

- Weitere Informationen finden Sie im AWS SDK for JavaScript -Entwicklerhandbuch.
- Einzelheiten zur API finden Sie DisableAlarmActionsin der AWS SDK for JavaScript API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun disableActions(alarmName: String) {

    val request = DisableAlarmActionsRequest {
        alarmNames = listOf(alarmName)
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.disableAlarmActions(request)
        println("Successfully disabled actions on alarm $alarmName")
    }
}
```

- API-Details finden Sie DisableAlarmActionsin der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def enable_alarm_actions(self, alarm_name, enable):
        """
        Enables or disables actions on the specified alarm. Alarm actions can be
        used to send notifications or automate responses when an alarm enters a
        particular state.

        :param alarm_name: The name of the alarm.
        :param enable: When True, actions are enabled for the alarm. Otherwise,
 they
                       disabled.
        """
        try:
            alarm = self.cloudwatch_resource.Alarm(alarm_name)
            if enable:
                alarm.enable_actions()
            else:
                alarm.disable_actions()
            logger.info(
                "%s actions for alarm %s.",
                "Enabled" if enable else "Disabled",
                alarm_name,
            )
```

```
        except ClientError:
            logger.exception(
                "Couldn't %s actions alarm %s.",
                "enable" if enable else "disable",
                alarm_name,
            )
            raise
```

- Einzelheiten zur API finden Sie DisableAlarmActionsin AWS SDK for Python (Boto3) API Reference.

Ruby

SDK für Ruby

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
# Disables an alarm in Amazon CloudWatch.
#
# Prerequisites.
#
# - The alarm to disable.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param alarm_name [String] The name of the alarm to disable.
# @return [Boolean] true if the alarm was disabled; otherwise, false.
# @example
#   exit 1 unless alarm_actions_disabled?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'ObjectsInBucket'
#   )
def alarm_actions_disabled?(cloudwatch_client, alarm_name)
  cloudwatch_client.disable_alarm_actions(alarm_names: [alarm_name])
  return true
```

```
rescue StandardError => e
  puts "Error disabling alarm actions: #{e.message}"
  return false
end

# Example usage:
def run_me
  alarm_name = "ObjectsInBucket"
  alarm_description = "Objects exist in this bucket for more than 1 day."
  metric_name = "NumberOfObjects"
  # Notify this Amazon Simple Notification Service (Amazon SNS) topic when
  # the alarm transitions to the ALARM state.
  alarm_actions = ["arn:aws:sns:us-
east-1:111111111111:Default_CloudWatch_Alarms_Topic"]
  namespace = "AWS/S3"
  statistic = "Average"
  dimensions = [
    {
      name: "BucketName",
      value: "doc-example-bucket"
    },
    {
      name: "StorageType",
      value: "AllStorageTypes"
    }
  ]
  period = 86_400 # Daily (24 hours * 60 minutes * 60 seconds = 86400 seconds).
  unit = "Count"
  evaluation_periods = 1 # More than one day.
  threshold = 1 # One object.
  comparison_operator = "GreaterThanThreshold" # More than one object.
  # Replace us-west-2 with the AWS Region you're using for Amazon CloudWatch.
  region = "us-east-1"

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)

  if alarm_created_or_updated?(
    cloudwatch_client,
    alarm_name,
    alarm_description,
    metric_name,
    alarm_actions,
    namespace,
    statistic,
```

```
    dimensions,
    period,
    unit,
    evaluation_periods,
    threshold,
    comparison_operator
  )
    puts "Alarm '#{alarm_name}' created or updated."
  else
    puts "Could not create or update alarm '#{alarm_name}'."
  end

  if alarm_actions_disabled?(cloudwatch_client, alarm_name)
    puts "Alarm '#{alarm_name}' disabled."
  else
    puts "Could not disable alarm '#{alarm_name}'."
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

- Einzelheiten zur API finden Sie DisableAlarmActionsin der AWS SDK for Ruby API-Referenz.

SAP ABAP

SDK für SAP ABAP

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    "Disables actions on the specified alarm. "
    TRY.
        lo_cwt->disablealarmactions(
          it_alarmnames = it_alarm_names
        ).
```

```
        MESSAGE 'Alarm actions disabled.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Einzelheiten zur API finden Sie DisableAlarmActionsin der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`EnableAlarmActions`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`EnableAlarmActions`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Metriken und Alarme verwalten

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    /// <summary>
    /// Enable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
```

```
    public async Task<bool> EnableAlarmActions(List<string> alarmNames)
    {
        var enableAlarmActionsResult = await
_amazonCloudWatch.EnableAlarmActionsAsync(
            new EnableAlarmActionsRequest()
            {
                AlarmNames = alarmNames
            });

        return enableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Einzelheiten zur API finden Sie EnableAlarmActionsin der AWS SDK for .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Binden Sie die erforderlichen Dateien ein.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/EnableAlarmActionsRequest.h>
#include <aws/monitoring/model/PutMetricAlarmRequest.h>
#include <iostream>
```

Aktivieren von Alarmaktionen

```
    Aws::CloudWatch::CloudWatchClient cw;
    Aws::CloudWatch::Model::PutMetricAlarmRequest request;
    request.SetAlarmName(alarm_name);
    request.SetComparisonOperator(
        Aws::CloudWatch::Model::ComparisonOperator::GreaterThanThreshold);
```

```
    request.SetEvaluationPeriods(1);
    request.SetMetricName("CPUUtilization");
    request.SetNamespace("AWS/EC2");
    request.SetPeriod(60);
    request.SetStatistic(Aws::CloudWatch::Model::Statistic::Average);
    request.SetThreshold(70.0);
    request.SetActionsEnabled(false);
    request.SetAlarmDescription("Alarm when server CPU exceeds 70%");
    request.SetUnit(Aws::CloudWatch::Model::StandardUnit::Seconds);
    request.AddAlarmActions(actionArn);

    Aws::CloudWatch::Model::Dimension dimension;
    dimension.SetName("InstanceId");
    dimension.SetValue(instanceId);
    request.AddDimensions(dimension);

    auto outcome = cw.PutMetricAlarm(request);
    if (!outcome.IsSuccess())
    {
        std::cout << "Failed to create CloudWatch alarm:" <<
            outcome.GetError().GetMessage() << std::endl;
        return;
    }

    Aws::CloudWatch::Model::EnableAlarmActionsRequest enable_request;
    enable_request.AddAlarmNames(alarm_name);

    auto enable_outcome = cw.EnableAlarmActions(enable_request);
    if (!enable_outcome.IsSuccess())
    {
        std::cout << "Failed to enable alarm actions:" <<
            enable_outcome.GetError().GetMessage() << std::endl;
        return;
    }

    std::cout << "Successfully created alarm " << alarm_name <<
        " and enabled actions on it." << std::endl;
```

- Einzelheiten zur API finden Sie EnableAlarmActionsin der AWS SDK for C++ API-Referenz.

CLI

AWS CLI

So aktivieren Sie alle Aktionen für einen Alarm

Das folgende Beispiel verwendet den `enable-alarm-actions`-Befehl, um alle Aktionen für den Alarm mit dem Namen „myalarm“ zu aktivieren:

```
aws cloudwatch enable-alarm-actions --alarm-names myalarm
```

Wenn dieser Befehl erfolgreich war, kehrt er zur Eingabeaufforderung zurück.

- Einzelheiten zur API finden Sie EnableAlarmActionsin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import
 software.amazon.awssdk.services.cloudwatch.model.EnableAlarmActionsRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class EnableAlarmActions {
    public static void main(String[] args) {
```

```
        final String usage = """

                Usage:
                  <alarmName>

                Where:
                  alarmName - An alarm name to enable (for example, MyAlarm).
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alarm = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        enableActions(cw, alarm);
        cw.close();
    }

    public static void enableActions(CloudWatchClient cw, String alarm) {
        try {
            EnableAlarmActionsRequest request =
 EnableAlarmActionsRequest.builder()
                    .alarmNames(alarm)
                    .build();

            cw.enableAlarmActions(request);
            System.out.printf("Successfully enabled actions on alarm %s", alarm);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie EnableAlarmActionsin der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Importieren Sie das SDK- und Client-Module und rufen Sie die API auf.

```
import { EnableAlarmActionsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new EnableAlarmActionsCommand({
    AlarmNames: [process.env.CLOUDWATCH_ALARM_NAME], // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Erstellen Sie den Client in einem separaten Modul und exportieren Sie ihn.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Weitere Informationen finden Sie im AWS SDK for JavaScript -Entwicklerhandbuch.
- Einzelheiten zur API finden Sie EnableAlarmActionsin der AWS SDK for JavaScript API-Referenz.

SDK für JavaScript (v2)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Importieren Sie das SDK- und Client-Module und rufen Sie die API auf.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  AlarmName: "Web_Server_CPU_Utilization",
  ComparisonOperator: "GreaterThanThreshold",
  EvaluationPeriods: 1,
  MetricName: "CPUUtilization",
  Namespace: "AWS/EC2",
  Period: 60,
  Statistic: "Average",
  Threshold: 70.0,
  ActionsEnabled: true,
  AlarmActions: ["ACTION_ARN"],
  AlarmDescription: "Alarm when server CPU exceeds 70%",
  Dimensions: [
    {
      Name: "InstanceId",
      Value: "INSTANCE_ID",
    },
  ],
  Unit: "Percent",
};
```

```
cw.putMetricAlarm(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Alarm action added", data);
    var paramsEnableAlarmAction = {
      AlarmNames: [params.AlarmName],
    };
    cw.enableAlarmActions(paramsEnableAlarmAction, function (err, data) {
      if (err) {
        console.log("Error", err);
      } else {
        console.log("Alarm action enabled", data);
      }
    });
  }
});
```

- Weitere Informationen finden Sie im AWS SDK for JavaScript -Entwicklerhandbuch.
- Einzelheiten zur API finden Sie EnableAlarmActionsin der AWS SDK for JavaScript API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun enableActions(alarm: String) {

    val request = EnableAlarmActionsRequest {
        alarmNames = listOf(alarm)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.enableAlarmActions(request)
```

```
        println("Successfully enabled actions on alarm $alarm")
    }
}
```

- API-Details finden Sie EnableAlarmActionsin der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def enable_alarm_actions(self, alarm_name, enable):
        """
        Enables or disables actions on the specified alarm. Alarm actions can be
        used to send notifications or automate responses when an alarm enters a
        particular state.

        :param alarm_name: The name of the alarm.
        :param enable: When True, actions are enabled for the alarm. Otherwise,
 they
                       disabled.
        """
        try:
            alarm = self.cloudwatch_resource.Alarm(alarm_name)
            if enable:
                alarm.enable_actions()
```

```
            else:
                alarm.disable_actions()
            logger.info(
                "%s actions for alarm %s.",
                "Enabled" if enable else "Disabled",
                alarm_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't %s actions alarm %s.",
                "enable" if enable else "disable",
                alarm_name,
            )
            raise
```

- Einzelheiten zur API finden Sie EnableAlarmActionsin AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    "Enable actions on the specified alarm."
    TRY.
        lo_cwt->enablealarmactions(
          it_alarmnames = it_alarm_names
        ).
        MESSAGE 'Alarm actions enabled.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
```

```
    ENDTRY.
```

- Einzelheiten zur API finden Sie EnableAlarmActionsin der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetDashboard`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetDashboard`.

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
/// <summary>
/// Get information on a dashboard.
/// </summary>
/// <param name="dashboardName">The name of the dashboard.</param>
/// <returns>A JSON object with dashboard information.</returns>
public async Task<string> GetDashboard(string dashboardName)
{
    var dashboardResponse = await _amazonCloudWatch.GetDashboardAsync(
        new GetDashboardRequest()
        {
            DashboardName = dashboardName
        });

    return dashboardResponse.DashboardBody;
}
```

```
```

- Einzelheiten zur API finden Sie GetDashboardin der AWS SDK for .NET API-Referenz.

PowerShell

Tools für PowerShell

Beispiel 1: Gibt den Hauptteil des angegebenen Dashboards zurück.

```
Get-CWDashboard -DashboardName Dashboard1
```

Ausgabe:

```
DashboardArn                                                DashboardBody
------------                                                -------------
arn:aws:cloudwatch::123456789012:dashboard/Dashboard1 {...
```

- Einzelheiten zur API finden Sie unter GetDashboard AWS Tools for PowerShellCmdlet-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwendung CloudWatch mit einem SDK AWS Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetMetricData`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetMetricData`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erste Schritte mit CloudWatch-Metriken, -Dashboards und -Alarmen

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    /// <summary>
    /// Get data for CloudWatch metrics.
    /// </summary>
    /// <param name="minutesOfData">The number of minutes of data to include.</
param>
    /// <param name="useDescendingTime">True to return the data descending by
 time.</param>
    /// <param name="endDateUtc">The end date for the data, in UTC.</param>
    /// <param name="maxDataPoints">The maximum data points to include.</param>
    /// <param name="dataQueries">Optional data queries to include.</param>
    /// <returns>A list of the requested metric data.</returns>
    public async Task<List<MetricDataResult>> GetMetricData(int minutesOfData,
 bool useDescendingTime, DateTime? endDateUtc = null,
        int maxDataPoints = 0, List<MetricDataQuery>? dataQueries = null)
    {
        var metricData = new List<MetricDataResult>();
        // If no end time is provided, use the current time for the end time.
        endDateUtc ??= DateTime.UtcNow;
        var timeZoneOffset =
 TimeZoneInfo.Local.GetUtcOffset(endDateUtc.Value.ToLocalTime());
        var startTimeUtc = endDateUtc.Value.AddMinutes(-minutesOfData);
        // The timezone string should be in the format +0000, so use the timezone
 offset to format it correctly.
        var timeZoneString = $"{timeZoneOffset.Hours:D2}
{timeZoneOffset.Minutes:D2}";
        var paginatedMetricData = _amazonCloudWatch.Paginators.GetMetricData(
            new GetMetricDataRequest()
            {
                StartTimeUtc = startTimeUtc,
                EndTimeUtc = endDateUtc.Value,
                LabelOptions = new LabelOptions { Timezone = timeZoneString },
```

```
                ScanBy = useDescendingTime ? ScanBy.TimestampDescending :
ScanBy.TimestampAscending,
                MaxDatapoints = maxDataPoints,
                MetricDataQueries = dataQueries,
            });

        await foreach (var data in paginatedMetricData.MetricDataResults)
        {
            metricData.Add(data);
        }
        return metricData;
    }
```

- Einzelheiten zur API finden Sie GetMetricDatain der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    public static void getCustomMetricData(CloudWatchClient cw, String fileName)
{
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            // Set the date.
            Instant nowDate = Instant.now();
```

```
            long hours = 1;
            long minutes = 30;
            Instant date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(minutes,
                    ChronoUnit.MINUTES);

            Metric met = Metric.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .build();

            MetricStat metStat = MetricStat.builder()
                    .stat("Maximum")
                    .period(1)
                    .metric(met)
                    .build();

            MetricDataQuery dataQUery = MetricDataQuery.builder()
                    .metricStat(metStat)
                    .id("foo2")
                    .returnData(true)
                    .build();

            List<MetricDataQuery> dq = new ArrayList<>();
            dq.add(dataQUery);

            GetMetricDataRequest getMetReq = GetMetricDataRequest.builder()
                    .maxDatapoints(10)
                    .scanBy(ScanBy.TIMESTAMP_DESCENDING)
                    .startTime(nowDate)
                    .endTime(date2)
                    .metricDataQueries(dq)
                    .build();

            GetMetricDataResponse response = cw.getMetricData(getMetReq);
            List<MetricDataResult> data = response.metricDataResults();
            for (MetricDataResult item : data) {
                System.out.println("The label is " + item.label());
                System.out.println("The status code is " +
item.statusCode().toString());
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
```

```
        }
    }
```

- Einzelheiten zur API finden Sie GetMetricDatain der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun getCustomMetricData(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set the date.
    val nowDate = Instant.now()
    val hours: Long = 1
    val minutes: Long = 30
    val date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(
        minutes,
        ChronoUnit.MINUTES
    )

    val met = Metric {
        metricName = customMetricName
        namespace = customMetricNamespace
    }

    val metStat = MetricStat {
        stat = "Maximum"
        period = 1
        metric = met
```

```
    }

    val dataQUery = MetricDataQuery {
        metricStat = metStat
        id = "foo2"
        returnData = true
    }

    val dq = ArrayList<MetricDataQuery>()
    dq.add(dataQUery)
    val getMetReq = GetMetricDataRequest {
        maxDatapoints = 10
        scanBy = ScanBy.TimestampDescending
        startTime = aws.smithy.kotlin.runtime.time.Instant(nowDate)
        endTime = aws.smithy.kotlin.runtime.time.Instant(date2)
        metricDataQueries = dq
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricData(getMetReq)
        response.metricDataResults?.forEach { item ->
            println("The label is ${item.label}")
            println("The status code is ${item.statusCode}")
        }
    }
}
```

- API-Details finden Sie GetMetricDatain der API-Referenz zum AWS SDK für Kotlin.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetMetricStatistics`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetMetricStatistics`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- Erste Schritte mit CloudWatch-Metriken, -Dashboards und -Alarmen

- Metriken und Alarme verwalten

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    /// <summary>
    /// Get billing statistics using a call to a wrapper class.
    /// </summary>
    /// <returns>A collection of billing statistics.</returns>
    private static async Task<List<Datapoint>> SetupBillingStatistics()
    {
        // Make a request for EstimatedCharges with a period of one day for the
past seven days.
        var billingStatistics = await _cloudWatchWrapper.GetMetricStatistics(
            "AWS/Billing",
            "EstimatedCharges",
            new List<string>() { "Maximum" },
            new List<Dimension>() { new Dimension { Name = "Currency", Value =
"USD" } },
            7,
            86400);

        billingStatistics = billingStatistics.OrderBy(n => n.Timestamp).ToList();

        return billingStatistics;
    }

    /// <summary>
    /// Wrapper to get statistics for a specific CloudWatch metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <param name="statistics">The list of statistics to include.</param>
    /// <param name="dimensions">The list of dimensions to include.</param>
    /// <param name="days">The number of days in the past to include.</param>
```

```
    /// <param name="period">The period for the data.</param>
    /// <returns>A list of DataPoint objects for the statistics.</returns>
    public async Task<List<Datapoint>> GetMetricStatistics(string
metricNamespace,
        string metricName, List<string> statistics, List<Dimension> dimensions,
int days, int period)
    {
        var metricStatistics = await _amazonCloudWatch.GetMetricStatisticsAsync(
            new GetMetricStatisticsRequest()
            {
                Namespace = metricNamespace,
                MetricName = metricName,
                Dimensions = dimensions,
                Statistics = statistics,
                StartTimeUtc = DateTime.UtcNow.AddDays(-days),
                EndTimeUtc = DateTime.UtcNow,
                Period = period
            });

        return metricStatistics.Datapoints;
    }
```

- Einzelheiten zur API finden Sie GetMetricStatisticsin der AWS SDK for .NET API-Referenz.

CLI

AWS CLI

So rufen Sie die CPU-Auslastung pro EC2-Instance ab

Im folgenden Beispiel wird der `get-metric-statistics`-Befehl verwendet, um die CPU-Auslastung für eine EC2-Instance mit der ID i-abcdef abzurufen.

```
aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-time
 2014-04-08T23:18:00Z --end-time 2014-04-09T23:18:00Z --period 3600 --namespace
 AWS/EC2 --statistics Maximum --dimensions Name=InstanceId,Value=i-abcdef
```

Ausgabe:

```
{
    "Datapoints": [
```

```
        {
            "Timestamp": "2014-04-09T11:18:00Z",
            "Maximum": 44.79,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T20:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T19:18:00Z",
            "Maximum": 50.85,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T09:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T03:18:00Z",
            "Maximum": 76.84,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T21:18:00Z",
            "Maximum": 48.96,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T14:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T08:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T16:18:00Z",
            "Maximum": 45.55,
            "Unit": "Percent"
```

```
        },
        {
            "Timestamp": "2014-04-09T06:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T13:18:00Z",
            "Maximum": 45.08,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T05:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T18:18:00Z",
            "Maximum": 46.88,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T17:18:00Z",
            "Maximum": 52.08,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T07:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T02:18:00Z",
            "Maximum": 51.23,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T12:18:00Z",
            "Maximum": 47.67,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-08T23:18:00Z",
            "Maximum": 46.88,
```

```
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T10:18:00Z",
            "Maximum": 51.91,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T04:18:00Z",
            "Maximum": 47.13,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T15:18:00Z",
            "Maximum": 48.96,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T00:18:00Z",
            "Maximum": 48.16,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T01:18:00Z",
            "Maximum": 49.18,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```

Angeben mehrerer Dimensionen

Das folgende Beispiel zeigt, wie mehrere Dimensionen angegeben werden können. Jede Dimension wird als Name/Wert-Paar mit einem Komma zwischen dem Namen und dem Wert angegeben. Mehrere Dimensionen sind durch ein Leerzeichen getrennt. Wenn eine einzelne Metrik mehrere Dimensionen enthält, müssen Sie für jede definierte Dimension einen Wert angeben.

Weitere Beispiele für die Verwendung des `get-metric-statistics` Befehls finden Sie unter Get Statistics for a Metric im Amazon CloudWatch Developer Guide.

```
aws cloudwatch get-metric-statistics --metric-name Buffers --
namespace MyNameSpace --dimensions Name=InstanceID,Value=i-abcdef
 Name=InstanceType,Value=m1.small --start-time 2016-10-15T04:00:00Z --end-time
 2016-10-19T07:00:00Z --statistics Average --period 60
```

- Einzelheiten zur API finden Sie GetMetricStatisticsunter AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
   public static void getAndDisplayMetricStatistics(CloudWatchClient cw, String
nameSpace, String metVal,
           String metricOption, String date, Dimension myDimension) {
       try {
           Instant start = Instant.parse(date);
           Instant endDate = Instant.now();

           GetMetricStatisticsRequest statisticsRequest =
GetMetricStatisticsRequest.builder()
                   .endTime(endDate)
                   .startTime(start)
                   .dimensions(myDimension)
                   .metricName(metVal)
                   .namespace(nameSpace)
                   .period(86400)
                   .statistics(Statistic.fromValue(metricOption))
                   .build();

           GetMetricStatisticsResponse response =
cw.getMetricStatistics(statisticsRequest);
           List<Datapoint> data = response.datapoints();
           if (!data.isEmpty()) {
               for (Datapoint datapoint : data) {
                   System.out
```

```
                            .println("Timestamp: " + datapoint.timestamp() + "
 Maximum value: " + datapoint.maximum());
                }
            } else {
                System.out.println("The returned data list is empty");
            }

        } catch (CloudWatchException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie GetMetricStatisticsin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun getAndDisplayMetricStatistics(nameSpaceVal: String, metVal: String,
 metricOption: String, date: String, myDimension: Dimension) {
    val start = Instant.parse(date)
    val endDate = Instant.now()
    val statisticsRequest = GetMetricStatisticsRequest {
        endTime = aws.smithy.kotlin.runtime.time.Instant(endDate)
        startTime = aws.smithy.kotlin.runtime.time.Instant(start)
        dimensions = listOf(myDimension)
        metricName = metVal
        namespace = nameSpaceVal
        period = 86400
        statistics = listOf(Statistic.fromValue(metricOption))
    }
```

```
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricStatistics(statisticsRequest)
        val data = response.datapoints
        if (data != null) {
            if (data.isNotEmpty()) {
                for (datapoint in data) {
                    println("Timestamp: ${datapoint.timestamp} Maximum value:
 ${datapoint.maximum}")
                }
            } else {
                println("The returned data list is empty")
            }
        }
    }
}
```

- API-Details finden Sie GetMetricStatisticsin der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def get_metric_statistics(self, namespace, name, start, end, period,
 stat_types):
        """
```

```
        Gets statistics for a metric within a specified time span. Metrics are
grouped
        into the specified period.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param start: The UTC start time of the time span to retrieve.
        :param end: The UTC end time of the time span to retrieve.
        :param period: The period, in seconds, in which to group metrics. The
period
                       must match the granularity of the metric, which depends on
                       the metric's age. For example, metrics that are older than
                       three hours have a one-minute granularity, so the period
must
                       be at least 60 and must be a multiple of 60.
        :param stat_types: The type of statistics to retrieve, such as average
value
                           or maximum value.
        :return: The retrieved statistics for the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            stats = metric.get_statistics(
                StartTime=start, EndTime=end, Period=period,
Statistics=stat_types
            )
            logger.info(
                "Got %s statistics for %s.", len(stats["Datapoints"]),
stats["Label"]
            )
        except ClientError:
            logger.exception("Couldn't get statistics for %s.%s.", namespace,
name)
            raise
        else:
            return stats
```

- Einzelheiten zur API finden Sie GetMetricStatisticsin AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwendung CloudWatch mit einem SDK AWS Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetMetricWidgetImage`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`GetMetricWidgetImage`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erste Schritte mit CloudWatch-Metriken, -Dashboards und -Alarmen

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    /// <summary>
    /// Get an image for a metric graphed over time.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metric">The name of the metric.</param>
    /// <param name="stat">The name of the stat to chart.</param>
    /// <param name="period">The period to use for the chart.</param>
    /// <returns>A memory stream for the chart image.</returns>
    public async Task<MemoryStream> GetTimeSeriesMetricImage(string
metricNamespace, string metric, string stat, int period)
    {
        var metricImageWidget = new
        {
            title = "Example Metric Graph",
            view = "timeSeries",
            stacked = false,
            period = period,
```

```
            width = 1400,
            height = 600,
            metrics = new List<List<object>>
                { new() { metricNamespace, metric, new { stat } } }
        };

        var metricImageWidgetString =
JsonSerializer.Serialize(metricImageWidget);
        var imageResponse = await _amazonCloudWatch.GetMetricWidgetImageAsync(
            new GetMetricWidgetImageRequest()
            {
                MetricWidget = metricImageWidgetString
            });

        return imageResponse.MetricWidgetImage;
    }

    /// <summary>
    /// Save a metric image to a file.
    /// </summary>
    /// <param name="memoryStream">The MemoryStream for the metric image.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The path to the file.</returns>
    public string SaveMetricImage(MemoryStream memoryStream, string metricName)
    {
        var metricFileName = $"{metricName}_{DateTime.Now.Ticks}.png";
        using var sr = new StreamReader(memoryStream);
        // Writes the memory stream to a file.
        File.WriteAllBytes(metricFileName, memoryStream.ToArray());
        var filePath = Path.Join(AppDomain.CurrentDomain.BaseDirectory,
            metricFileName);
        return filePath;
    }
```

- Einzelheiten zur API finden Sie GetMetricWidgetImagein der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
   public static void getAndOpenMetricImage(CloudWatchClient cw, String
fileName) {
       System.out.println("Getting Image data for custom metric.");
       try {
           String myJSON = "{\n" +
                   "  \"title\": \"Example Metric Graph\",\n" +
                   "  \"view\": \"timeSeries\",\n" +
                   "  \"stacked \": false,\n" +
                   "  \"period\": 10,\n" +
                   "  \"width\": 1400,\n" +
                   "  \"height\": 600,\n" +
                   "  \"metrics\": [\n" +
                   "    [\n" +
                   "      \"AWS/Billing\",\n" +
                   "      \"EstimatedCharges\",\n" +
                   "      \"Currency\",\n" +
                   "      \"USD\"\n" +
                   "    ]\n" +
                   "  ]\n" +
                   "}";

           GetMetricWidgetImageRequest imageRequest =
GetMetricWidgetImageRequest.builder()
                   .metricWidget(myJSON)
                   .build();

           GetMetricWidgetImageResponse response =
cw.getMetricWidgetImage(imageRequest);
           SdkBytes sdkBytes = response.metricWidgetImage();
           byte[] bytes = sdkBytes.asByteArray();
           File outputFile = new File(fileName);
```

```
            try (FileOutputStream outputStream = new
 FileOutputStream(outputFile)) {
                outputStream.write(bytes);
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie GetMetricWidgetImagein der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun getAndOpenMetricImage(fileName: String) {
    println("Getting Image data for custom metric.")
    val myJSON = """{
        "title": "Example Metric Graph",
        "view": "timeSeries",
        "stacked ": false,
        "period": 10,
        "width": 1400,
        "height": 600,
        "metrics": [
            [
            "AWS/Billing",
            "EstimatedCharges",
            "Currency",
            "USD"
            ]
```

```
        ]
        }"""

    val imageRequest = GetMetricWidgetImageRequest {
        metricWidget = myJSON
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricWidgetImage(imageRequest)
        val bytes = response.metricWidgetImage
        if (bytes != null) {
            File(fileName).writeBytes(bytes)
        }
    }
    println("You have successfully written data to $fileName")
}
```

- API-Details finden Sie GetMetricWidgetImagein der API-Referenz zum AWS SDK für Kotlin.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`ListDashboards`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`ListDashboards`.

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    /// <summary>
    /// Get a list of dashboards.
    /// </summary>
```

```
    /// <returns>A list of DashboardEntry objects.</returns>
    public async Task<List<DashboardEntry>> ListDashboards()
    {
        var results = new List<DashboardEntry>();
        var paginateDashboards = _amazonCloudWatch.Paginators.ListDashboards(
            new ListDashboardsRequest());
        // Get the entire list using the paginator.
        await foreach (var data in paginateDashboards.DashboardEntries)
        {
            results.Add(data);
        }

        return results;
    }
```

- Einzelheiten zur API finden Sie ListDashboardsin der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    public static void listDashboards(CloudWatchClient cw) {
        try {
            ListDashboardsIterable listRes = cw.listDashboardsPaginator();
            listRes.stream()
                    .flatMap(r -> r.dashboardEntries().stream())
                    .forEach(entry -> {
                        System.out.println("Dashboard name is: " +
entry.dashboardName());
                        System.out.println("Dashboard ARN is: " +
entry.dashboardArn());
                    });

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
```

```
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie ListDashboardsin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun listDashboards() {
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.listDashboardsPaginated({})
            .transform { it.dashboardEntries?.forEach { obj -> emit(obj) } }
            .collect { obj ->
                println("Name is ${obj.dashboardName}")
                println("Dashboard ARN is ${obj.dashboardArn}")
            }
    }
}
```

- API-Details finden Sie ListDashboardsin der API-Referenz zum AWS SDK für Kotlin.

PowerShell

Tools für PowerShell

Beispiel 1: Gibt die Sammlung von Dashboards für Ihr Konto zurück.

```
Get-CWDashboardList
```

Ausgabe:

```
DashboardArn DashboardName LastModified        Size
------------ ------------- ------------        ----
arn:...      Dashboard1    7/6/2017 8:14:15 PM 252
```

Beispiel 2: Gibt die Sammlung von Dashboards für Ihr Konto zurück, deren Namen mit dem Präfix „dev“ beginnen.

```
Get-CWDashboardList -DashboardNamePrefix dev
```

- Einzelheiten zur API finden Sie unter ListDashboardsCmdlet-Referenz.AWS Tools for PowerShell

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwendung CloudWatch mit einem SDK AWS Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`ListMetrics`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`ListMetrics`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- Erste Schritte mit CloudWatch-Metriken, -Dashboards und -Alarmen
- Metriken und Alarme verwalten

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    /// <summary>
```

```
    /// List metrics available, optionally within a namespace.
    /// </summary>
    /// <param name="metricNamespace">Optional CloudWatch namespace to use when
listing metrics.</param>
    /// <param name="filter">Optional dimension filter.</param>
    /// <param name="metricName">Optional metric name filter.</param>
    /// <returns>The list of metrics.</returns>
    public async Task<List<Metric>> ListMetrics(string? metricNamespace = null,
DimensionFilter? filter = null, string? metricName = null)
    {
        var results = new List<Metric>();
        var paginateMetrics = _amazonCloudWatch.Paginators.ListMetrics(
            new ListMetricsRequest
            {
                Namespace = metricNamespace,
                Dimensions = filter != null ? new List<DimensionFilter>
{ filter } : null,
                MetricName = metricName
            });
        // Get the entire list using the paginator.
        await foreach (var metric in paginateMetrics.Metrics)
        {
            results.Add(metric);
        }

        return results;
    }
```

- Einzelheiten zur API finden Sie ListMetricsin der AWS SDK for .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Binden Sie die erforderlichen Dateien ein.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/ListMetricsRequest.h>
#include <aws/monitoring/model/ListMetricsResult.h>
#include <iomanip>
#include <iostream>
```

Listen Sie die Metriken auf.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::ListMetricsRequest request;

        if (argc > 1)
        {
            request.SetMetricName(argv[1]);
        }

        if (argc > 2)
        {
            request.SetNamespace(argv[2]);
        }

        bool done = false;
        bool header = false;
        while (!done)
        {
            auto outcome = cw.ListMetrics(request);
            if (!outcome.IsSuccess())
            {
                std::cout << "Failed to list CloudWatch metrics:" <<
                    outcome.GetError().GetMessage() << std::endl;
                break;
            }

            if (!header)
            {
                std::cout << std::left << std::setw(48) << "MetricName" <<
                    std::setw(32) << "Namespace" << "DimensionNameValuePairs" <<
                    std::endl;
                header = true;
            }
```

```
            const auto &metrics = outcome.GetResult().GetMetrics();
            for (const auto &metric : metrics)
            {
                std::cout << std::left << std::setw(48) <<
                    metric.GetMetricName() << std::setw(32) <<
                    metric.GetNamespace();
                const auto &dimensions = metric.GetDimensions();
                for (auto iter = dimensions.cbegin();
                    iter != dimensions.cend(); ++iter)
                {
                    const auto &dimkv = *iter;
                    std::cout << dimkv.GetName() << " = " << dimkv.GetValue();
                    if (iter + 1 != dimensions.cend())
                    {
                        std::cout << ", ";
                    }
                }
                std::cout << std::endl;
            }

            const auto &next_token = outcome.GetResult().GetNextToken();
            request.SetNextToken(next_token);
            done = next_token.empty();
        }
```

- Einzelheiten zur API finden Sie ListMetricsin der AWS SDK for C++ API-Referenz.

CLI

AWS CLI

So listen Sie die Metriken für Amazon SNS auf

Im folgenden `list-metrics`-Beispiel werden die Metriken für Amazon SNS angezeigt.

```
aws cloudwatch list-metrics \
    --namespace "AWS/SNS"
```

Ausgabe:

```
{
    "Metrics": [
```

```
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "PublishSize"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "PublishSize"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfNotificationsFailed"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfNotificationsDelivered"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
```

```
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfMessagesPublished"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfMessagesPublished"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfNotificationsDelivered"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfNotificationsFailed"
        }
    ]
}
```

- Einzelheiten zur API finden Sie ListMetricsin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsRequest;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsResponse;
import software.amazon.awssdk.services.cloudwatch.model.Metric;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListMetrics {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <namespace>\s

                Where:
                  namespace - The namespace to filter against (for example, AWS/
EC2).\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        String namespace = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        listMets(cw, namespace);
        cw.close();
    }

    public static void listMets(CloudWatchClient cw, String namespace) {
        boolean done = false;
        String nextToken = null;

        try {
            while (!done) {

                ListMetricsResponse response;
                if (nextToken == null) {
                    ListMetricsRequest request = ListMetricsRequest.builder()
                            .namespace(namespace)
                            .build();

                    response = cw.listMetrics(request);
                } else {
                    ListMetricsRequest request = ListMetricsRequest.builder()
                            .namespace(namespace)
                            .nextToken(nextToken)
                            .build();

                    response = cw.listMetrics(request);
                }

                for (Metric metric : response.metrics()) {
                    System.out.printf("Retrieved metric %s",
metric.metricName());
                    System.out.println();
                }

                if (response.nextToken() == null) {
                    done = true;
                } else {
                    nextToken = response.nextToken();
```

```
                }
            }

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie ListMetricsin der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Importieren Sie das SDK- und Client-Module und rufen Sie die API auf.

```
import { ListMetricsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

export const main = () => {
  // Use the AWS console to see available namespaces and metric names. Custom
 metrics can also be created.
  // https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
viewing_metrics_with_cloudwatch.html
  const command = new ListMetricsCommand({
    Dimensions: [
      {
        Name: "LogGroupName",
      },
    ],
    MetricName: "IncomingLogEvents",
    Namespace: "AWS/Logs",
  });
```

```
  return client.send(command);
};
```

Erstellen Sie den Client in einem separaten Modul und exportieren Sie ihn.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Weitere Informationen finden Sie im AWS SDK for JavaScript -Entwicklerhandbuch.
- Einzelheiten zur API finden Sie ListMetricsin der AWS SDK for JavaScript API-Referenz.

SDK für JavaScript (v2)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  Dimensions: [
    {
      Name: "LogGroupName" /* required */,
    },
  ],
  MetricName: "IncomingLogEvents",
  Namespace: "AWS/Logs",
};
```

```
cw.listMetrics(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Metrics", JSON.stringify(data.Metrics));
  }
});
```

- Weitere Informationen finden Sie im AWS SDK for JavaScript -Entwicklerhandbuch.
- Einzelheiten zur API finden Sie ListMetricsin der AWS SDK for JavaScript API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun listMets(namespaceVal: String?): ArrayList<String>? {
    val metList = ArrayList<String>()
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val reponse = cwClient.listMetrics(request)
        reponse.metrics?.forEach { metrics ->
            val data = metrics.metricName
            if (!metList.contains(data)) {
                metList.add(data!!)
            }
        }
    }
    return metList
}
```

- API-Details finden Sie ListMetricsin der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def list_metrics(self, namespace, name, recent=False):
        """
        Gets the metrics within a namespace that have the specified name.
        If the metric has no dimensions, a single metric is returned.
        Otherwise, metrics for all dimensions are returned.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param recent: When True, only metrics that have been active in the last
                       three hours are returned.
        :return: An iterator that yields the retrieved metrics.
        """
        try:
            kwargs = {"Namespace": namespace, "MetricName": name}
            if recent:
                kwargs["RecentlyActive"] = "PT3H"  # List past 3 hours only
            metric_iter = self.cloudwatch_resource.metrics.filter(**kwargs)
            logger.info("Got metrics for %s.%s.", namespace, name)
        except ClientError:
            logger.exception("Couldn't get metrics for %s.%s.", namespace, name)
            raise
        else:
```

```
            return metric_iter
```

- Einzelheiten zur API finden Sie ListMetricsin AWS SDK for Python (Boto3) API Reference.

Ruby

SDK für Ruby

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
# Lists available metrics for a metric namespace in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param metric_namespace [String] The namespace of the metric.
# @example
#   list_metrics_for_namespace(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'SITE/TRAFFIC'
#   )
def list_metrics_for_namespace(cloudwatch_client, metric_namespace)
  response = cloudwatch_client.list_metrics(namespace: metric_namespace)

  if response.metrics.count.positive?
    response.metrics.each do |metric|
      puts "  Metric name: #{metric.metric_name}"
      if metric.dimensions.count.positive?
        puts "    Dimensions:"
        metric.dimensions.each do |dimension|
          puts "      Name: #{dimension.name}, Value: #{dimension.value}"
        end
      else
        puts "No dimensions found."
      end
    end
  else
```

```
    puts "No metrics found for namespace '#{metric_namespace}'. " \
      "Note that it could take up to 15 minutes for recently-added metrics " \
      "to become available."
  end
end

# Example usage:
def run_me
  metric_namespace = "SITE/TRAFFIC"
  # Replace us-west-2 with the AWS Region you're using for Amazon CloudWatch.
  region = "us-east-1"

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)

  # Add three datapoints.
  puts "Continuing..." unless datapoint_added_to_metric?(
    cloudwatch_client,
    metric_namespace,
    "UniqueVisitors",
    "SiteName",
    "example.com",
    5_885.0,
    "Count"
  )

  puts "Continuing..." unless datapoint_added_to_metric?(
    cloudwatch_client,
    metric_namespace,
    "UniqueVisits",
    "SiteName",
    "example.com",
    8_628.0,
    "Count"
  )

  puts "Continuing..." unless datapoint_added_to_metric?(
    cloudwatch_client,
    metric_namespace,
    "PageViews",
    "PageURL",
    "example.html",
    18_057.0,
    "Count"
  )
```

```
  puts "Metrics for namespace '#{metric_namespace}':"
  list_metrics_for_namespace(cloudwatch_client, metric_namespace)
end

run_me if $PROGRAM_NAME == __FILE__
```

- Einzelheiten zur API finden Sie ListMetricsin der AWS SDK for Ruby API-Referenz.

SAP ABAP

SDK für SAP ABAP

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
   "The following list-metrics example displays the metrics for Amazon
CloudWatch."
   TRY.
       oo_result = lo_cwt->listmetrics(            " oo_result is returned for
testing purposes. "
         iv_namespace = iv_namespace
       ).
       DATA(lt_metrics) = oo_result->get_metrics( ).
       MESSAGE 'Metrics retrieved.' TYPE 'I'.
     CATCH /aws1/cx_cwtinvparamvalueex .
       MESSAGE 'The specified argument was not valid.' TYPE 'E'.
   ENDTRY.
```

- Einzelheiten zur API finden Sie ListMetricsin der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **PutAnomalyDetector** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`PutAnomalyDetector`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erste Schritte mit CloudWatch-Metriken, -Dashboards und -Alarmen

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
   /// <summary>
   /// Add an anomaly detector for a single metric.
   /// </summary>
   /// <param name="anomalyDetector">A single metric anomaly detector.</param>
   /// <returns>True if successful.</returns>
   public async Task<bool> PutAnomalyDetector(SingleMetricAnomalyDetector
anomalyDetector)
   {
       var putAlarmDetectorResult = await
_amazonCloudWatch.PutAnomalyDetectorAsync(
           new PutAnomalyDetectorRequest()
           {
               SingleMetricAnomalyDetector = anomalyDetector
           });

       return putAlarmDetectorResult.HttpStatusCode == HttpStatusCode.OK;
   }
```

- Einzelheiten zur API finden Sie PutAnomalyDetectorin der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
   public static void addAnomalyDetector(CloudWatchClient cw, String fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();

           SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .stat("Maximum")
                   .build();

           PutAnomalyDetectorRequest anomalyDetectorRequest =
PutAnomalyDetectorRequest.builder()
                   .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                   .build();

           cw.putAnomalyDetector(anomalyDetectorRequest);
           System.out.println("Added anomaly detector for metric " +
customMetricName + ".");

       } catch (CloudWatchException | IOException e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
```

```
    }
```

- Einzelheiten zur API finden Sie PutAnomalyDetectorin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun addAnomalyDetector(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val anomalyDetectorRequest = PutAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putAnomalyDetector(anomalyDetectorRequest)
        println("Added anomaly detector for metric $customMetricName.")
    }
}
```

- API-Details finden Sie PutAnomalyDetectorin der API-Referenz zum AWS SDK für Kotlin.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **PutDashboard** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`PutDashboard`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erste Schritte mit CloudWatch-Metriken, -Dashboards und -Alarmen

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    /// <summary>
    /// Set up a dashboard using a call to the wrapper class.
    /// </summary>
    /// <param name="customMetricNamespace">The metric namespace.</param>
    /// <param name="customMetricName">The metric name.</param>
    /// <param name="dashboardName">The name of the dashboard.</param>
    /// <returns>A list of validation messages.</returns>
    private static async Task<List<DashboardValidationMessage>> SetupDashboard(
        string customMetricNamespace, string customMetricName, string
dashboardName)
    {
        // Get the dashboard model from configuration.
        var newDashboard = new DashboardModel();
        _configuration.GetSection("dashboardExampleBody").Bind(newDashboard);
```

```
        // Add a new metric to the dashboard.
        newDashboard.Widgets.Add(new Widget
        {
            Height = 8,
            Width = 8,
            Y = 8,
            X = 0,
            Type = "metric",
            Properties = new Properties
            {
                Metrics = new List<List<object>>
                    { new() { customMetricNamespace, customMetricName } },
                View = "timeSeries",
                Region = "us-east-1",
                Stat = "Sum",
                Period = 86400,
                YAxis = new YAxis { Left = new Left { Min = 0, Max = 100 } },
                Title = "Custom Metric Widget",
                LiveData = true,
                Sparkline = true,
                Trend = true,
                Stacked = false,
                SetPeriodToTimeRange = false
            }
        });

        var newDashboardString = JsonSerializer.Serialize(newDashboard,
            new JsonSerializerOptions
            { DefaultIgnoreCondition = JsonIgnoreCondition.WhenWritingNull });
        var validationMessages =
            await _cloudWatchWrapper.PutDashboard(dashboardName,
 newDashboardString);

        return validationMessages;
    }

    /// <summary>
    /// Wrapper to create or add to a dashboard with metrics.
    /// </summary>
    /// <param name="dashboardName">The name for the dashboard.</param>
    /// <param name="dashboardBody">The metric data in JSON for the dashboard.</
param>
    /// <returns>A list of validation messages for the dashboard.</returns>
```

```
    public async Task<List<DashboardValidationMessage>> PutDashboard(string
dashboardName,
        string dashboardBody)
    {
        // Updating a dashboard replaces all contents.
        // Best practice is to include a text widget indicating this dashboard
was created programmatically.
        var dashboardResponse = await _amazonCloudWatch.PutDashboardAsync(
            new PutDashboardRequest()
            {
                DashboardName = dashboardName,
                DashboardBody = dashboardBody
            });

        return dashboardResponse.DashboardValidationMessages;
    }
```

- Einzelheiten zur API finden Sie PutDashboardin der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    public static void createDashboardWithMetrics(CloudWatchClient cw, String
dashboardName, String fileName) {
        try {
            PutDashboardRequest dashboardRequest = PutDashboardRequest.builder()
                    .dashboardName(dashboardName)
                    .dashboardBody(readFileAsString(fileName))
                    .build();

            PutDashboardResponse response = cw.putDashboard(dashboardRequest);
            System.out.println(dashboardName + " was successfully created.");
```

```
            List<DashboardValidationMessage> messages =
 response.dashboardValidationMessages();
            if (messages.isEmpty()) {
                System.out.println("There are no messages in the new Dashboard");
            } else {
                for (DashboardValidationMessage message : messages) {
                    System.out.println("Message is: " + message.message());
                }
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie PutDashboardin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun createDashboardWithMetrics(dashboardNameVal: String, fileNameVal:
 String) {
    val dashboardRequest = PutDashboardRequest {
        dashboardName = dashboardNameVal
        dashboardBody = readFileAsString(fileNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.putDashboard(dashboardRequest)
        println("$dashboardNameVal was successfully created.")
        val messages = response.dashboardValidationMessages
        if (messages != null) {
            if (messages.isEmpty()) {
```

```
                println("There are no messages in the new Dashboard")
            } else {
                for (message in messages) {
                    println("Message is: ${message.message}")
                }
            }
        }
    }
}
```

- API-Details finden Sie PutDashboardin der API-Referenz zum AWS SDK für Kotlin.

PowerShell

Tools für PowerShell

Beispiel 1: Erstellt oder aktualisiert das Dashboard mit dem Namen 'Dashboard1', sodass es zwei Metrik-Widgets nebeneinander enthält.

```
$dashBody = @"
{
    "widgets":[
        {
             "type":"metric",
             "x":0,
             "y":0,
             "width":12,
             "height":6,
             "properties":{
                "metrics":[
                   [
                      "AWS/EC2",
                      "CPUUtilization",
                      "InstanceId",
                      "i-012345"
                   ]
                ],
                "period":300,
                "stat":"Average",
                "region":"us-east-1",
                "title":"EC2 Instance CPU"
             }
```

```
        },
        {
             "type":"metric",
             "x":12,
             "y":0,
             "width":12,
             "height":6,
             "properties":{
                "metrics":[
                   [
                      "AWS/S3",
                      "BucketSizeBytes",
                      "BucketName",
                      "MyBucketName"
                   ]
                ],
                "period":86400,
                "stat":"Maximum",
                "region":"us-east-1",
                "title":"MyBucketName bytes"
            }
        }
    ]
}
"@

Write-CWDashboard -DashboardName Dashboard1 -DashboardBody $dashBody
```

Beispiel 2: Erstellt oder aktualisiert das Dashboard und leitet den Inhalt, der das Dashboard beschreibt, über die Pipeline an das Cmdlet weiter.

```
$dashBody = @"
{
...
}
"@

$dashBody | Write-CWDashboard -DashboardName Dashboard1
```

- Einzelheiten zur API finden Sie unter PutDashboardCmdlet-Referenz.AWS Tools for PowerShell

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwendung CloudWatch mit einem SDK AWS Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`PutMetricAlarm`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`PutMetricAlarm`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- Erste Schritte mit Alarmen
- Erste Schritte mit CloudWatch-Metriken, -Dashboards und -Alarmen
- Metriken und Alarme verwalten

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    /// <summary>
    /// Add a metric alarm to send an email when the metric passes a threshold.
    /// </summary>
    /// <param name="alarmDescription">A description of the alarm.</param>
    /// <param name="alarmName">The name for the alarm.</param>
    /// <param name="comparison">The type of comparison to use.</param>
    /// <param name="metricName">The name of the metric for the alarm.</param>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="threshold">The threshold value for the alarm.</param>
    /// <param name="alarmActions">Optional actions to execute when in an alarm
state.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutMetricEmailAlarm(string alarmDescription, string
alarmName, ComparisonOperator comparison,
```

```
        string metricName, string metricNamespace, double threshold, List<string>
alarmActions = null!)
    {
        try
        {
            var putEmailAlarmResponse = await
_amazonCloudWatch.PutMetricAlarmAsync(
                new PutMetricAlarmRequest()
                {
                    AlarmActions = alarmActions,
                    AlarmDescription = alarmDescription,
                    AlarmName = alarmName,
                    ComparisonOperator = comparison,
                    Threshold = threshold,
                    Namespace = metricNamespace,
                    MetricName = metricName,
                    EvaluationPeriods = 1,
                    Period = 10,
                    Statistic = new Statistic("Maximum"),
                    DatapointsToAlarm = 1,
                    TreatMissingData = "ignore"
                });
            return putEmailAlarmResponse.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (LimitExceededException lex)
        {
            _logger.LogError(lex, $"Unable to add alarm {alarmName}. Alarm quota
has already been reached.");
        }

        return false;
    }

    /// <summary>
    /// Add specific email actions to a list of action strings for a CloudWatch
alarm.
    /// </summary>
    /// <param name="accountId">The AccountId for the alarm.</param>
    /// <param name="region">The region for the alarm.</param>
    /// <param name="emailTopicName">An Amazon Simple Notification Service (SNS)
topic for the alarm email.</param>
    /// <param name="alarmActions">Optional list of existing alarm actions to
append to.</param>
    /// <returns>A list of string actions for an alarm.</returns>
```

```
    public List<string> AddEmailAlarmAction(string accountId, string region,
        string emailTopicName, List<string>? alarmActions = null)
    {
        alarmActions ??= new List<string>();
        var snsAlarmAction = $"arn:aws:sns:{region}:{accountId}:
{emailTopicName}";
        alarmActions.Add(snsAlarmAction);
        return alarmActions;
    }
```

- Einzelheiten zur API finden Sie PutMetricAlarmin der AWS SDK for .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Binden Sie die erforderlichen Dateien ein.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/PutMetricAlarmRequest.h>
#include <iostream>
```

Erstellen Sie den Alarm, um die Metrik zu beobachten.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::PutMetricAlarmRequest request;
        request.SetAlarmName(alarm_name);
        request.SetComparisonOperator(
            Aws::CloudWatch::Model::ComparisonOperator::GreaterThanThreshold);
        request.SetEvaluationPeriods(1);
        request.SetMetricName("CPUUtilization");
        request.SetNamespace("AWS/EC2");
```

```
        request.SetPeriod(60);
        request.SetStatistic(Aws::CloudWatch::Model::Statistic::Average);
        request.SetThreshold(70.0);
        request.SetActionsEnabled(false);
        request.SetAlarmDescription("Alarm when server CPU exceeds 70%");
        request.SetUnit(Aws::CloudWatch::Model::StandardUnit::Seconds);

        Aws::CloudWatch::Model::Dimension dimension;
        dimension.SetName("InstanceId");
        dimension.SetValue(instanceId);

        request.AddDimensions(dimension);

        auto outcome = cw.PutMetricAlarm(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to create CloudWatch alarm:" <<
                outcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout << "Successfully created CloudWatch alarm " << alarm_name
                << std::endl;
        }
```

- Einzelheiten zur API finden Sie PutMetricAlarmin der AWS SDK for C++ API-Referenz.

CLI

AWS CLI

So senden Sie eine E-Mail-Nachricht von Amazon Simple Notification Service, wenn die CPU-Auslastung 70 % übersteigt

Im folgenden Beispiel wird der `put-metric-alarm`-Befehl verwendet, um eine E-Mail-Nachricht von Amazon Simple Notification Service zu senden, wenn die CPU-Auslastung 70 % übersteigt:

```
aws cloudwatch put-metric-alarm --alarm-name cpu-mon --alarm-description "Alarm
 when CPU exceeds 70 percent" --metric-name CPUUtilization --namespace AWS/
EC2 --statistic Average --period 300 --threshold 70 --comparison-operator
```

```
 GreaterThanThreshold  --dimensions "Name=InstanceId,Value=i-12345678" --
evaluation-periods 2 --alarm-actions arn:aws:sns:us-east-1:111122223333:MyTopic
 --unit Percent
```

Wenn dieser Befehl erfolgreich war, kehrt er zur Eingabeaufforderung zurück. Wenn ein Alarm mit demselben Namen bereits vorhanden ist, wird er durch den neuen Alarm überschrieben.

So geben Sie mehrere Dimensionen an

Das folgende Beispiel zeigt, wie mehrere Dimensionen angegeben werden können. Jede Dimension wird als Name/Wert-Paar mit einem Komma zwischen dem Namen und dem Wert angegeben. Mehrere Dimensionen werden durch ein Leerzeichen getrennt:

```
aws cloudwatch put-metric-alarm --alarm-name "Default_Test_Alarm3" --alarm-
description "The default example alarm" --namespace "CW EXAMPLE METRICS" --
metric-name Default_Test --statistic Average --period 60 --evaluation-periods 3
 --threshold 50 --comparison-operator GreaterThanOrEqualToThreshold --dimensions
 Name=key1,Value=value1 Name=key2,Value=value2
```

- Einzelheiten zur API finden Sie PutMetricAlarmin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    public static String createAlarm(CloudWatchClient cw, String fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
```

```
            String customMetricName =
rootNode.findValue("customMetricName").asText();
            String alarmName = rootNode.findValue("exampleAlarmName").asText();
            String emailTopic = rootNode.findValue("emailTopic").asText();
            String accountId = rootNode.findValue("accountId").asText();
            String region = rootNode.findValue("region").asText();

            // Create a List for alarm actions.
            List<String> alarmActions = new ArrayList<>();
            alarmActions.add("arn:aws:sns:" + region + ":" + accountId + ":" +
emailTopic);
            PutMetricAlarmRequest alarmRequest = PutMetricAlarmRequest.builder()
                    .alarmActions(alarmActions)
                    .alarmDescription("Example metric alarm")
                    .alarmName(alarmName)

.comparisonOperator(ComparisonOperator.GREATER_THAN_OR_EQUAL_TO_THRESHOLD)
                    .threshold(100.00)
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .evaluationPeriods(1)
                    .period(10)
                    .statistic("Maximum")
                    .datapointsToAlarm(1)
                    .treatMissingData("ignore")
                    .build();

            cw.putMetricAlarm(alarmRequest);
            System.out.println(alarmName + " was successfully created!");
            return alarmName;

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
        return "";
    }
```

- Einzelheiten zur API finden Sie PutMetricAlarmin der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Importieren Sie das SDK- und Client-Module und rufen Sie die API auf.

```
import { PutMetricAlarmCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  // This alarm triggers when CPUUtilization exceeds 70% for one minute.
  const command = new PutMetricAlarmCommand({
    AlarmName: process.env.CLOUDWATCH_ALARM_NAME, // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
    ComparisonOperator: "GreaterThanThreshold",
    EvaluationPeriods: 1,
    MetricName: "CPUUtilization",
    Namespace: "AWS/EC2",
    Period: 60,
    Statistic: "Average",
    Threshold: 70.0,
    ActionsEnabled: false,
    AlarmDescription: "Alarm when server CPU exceeds 70%",
    Dimensions: [
      {
        Name: "InstanceId",
        Value: process.env.EC2_INSTANCE_ID, // Set the value of EC_INSTANCE_ID to
 the Id of an existing Amazon EC2 instance.
      },
    ],
    Unit: "Percent",
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
```

```
  }
};

export default run();
```

Erstellen Sie den Client in einem separaten Modul und exportieren Sie ihn.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Weitere Informationen finden Sie im AWS SDK for JavaScript -Entwicklerhandbuch.
- Einzelheiten zur API finden Sie PutMetricAlarmin der AWS SDK for JavaScript API-Referenz.

SDK für JavaScript (v2)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  AlarmName: "Web_Server_CPU_Utilization",
  ComparisonOperator: "GreaterThanThreshold",
  EvaluationPeriods: 1,
  MetricName: "CPUUtilization",
  Namespace: "AWS/EC2",
  Period: 60,
  Statistic: "Average",
  Threshold: 70.0,
```

```
  ActionsEnabled: false,
  AlarmDescription: "Alarm when server CPU exceeds 70%",
  Dimensions: [
    {
      Name: "InstanceId",
      Value: "INSTANCE_ID",
    },
  ],
  Unit: "Percent",
};

cw.putMetricAlarm(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Weitere Informationen finden Sie im AWS SDK for JavaScript -Entwicklerhandbuch.
- Einzelheiten zur API finden Sie PutMetricAlarmin der AWS SDK for JavaScript API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun putMetricAlarm(alarmNameVal: String, instanceIdVal: String) {

    val dimensionOb = Dimension {
        name = "InstanceId"
        value = instanceIdVal
    }
```

```
    val request = PutMetricAlarmRequest {
        alarmName = alarmNameVal
        comparisonOperator = ComparisonOperator.GreaterThanThreshold
        evaluationPeriods = 1
        metricName = "CPUUtilization"
        namespace = "AWS/EC2"
        period = 60
        statistic = Statistic.fromValue("Average")
        threshold = 70.0
        actionsEnabled = false
        alarmDescription = "An Alarm created by the Kotlin SDK when server CPU
 utilization exceeds 70%"
        unit = StandardUnit.fromValue("Seconds")
        dimensions = listOf(dimensionOb)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricAlarm(request)
        println("Successfully created an alarm with name $alarmNameVal")
    }
}
```

- API-Details finden Sie PutMetricAlarmin der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
```

```
        self.cloudwatch_resource = cloudwatch_resource


    def create_metric_alarm(
        self,
        metric_namespace,
        metric_name,
        alarm_name,
        stat_type,
        period,
        eval_periods,
        threshold,
        comparison_op,
    ):
        """
        Creates an alarm that watches a metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        :param alarm_name: The name of the alarm.
        :param stat_type: The type of statistic the alarm watches.
        :param period: The period in which metric data are grouped to calculate
                       statistics.
        :param eval_periods: The number of periods that the metric must be over
the
                             alarm threshold before the alarm is set into an
alarmed
                             state.
        :param threshold: The threshold value to compare against the metric
statistic.
        :param comparison_op: The comparison operation used to compare the
threshold
                              against the metric.
        :return: The newly created alarm.
        """
        try:
            metric = self.cloudwatch_resource.Metric(metric_namespace,
metric_name)
            alarm = metric.put_alarm(
                AlarmName=alarm_name,
                Statistic=stat_type,
                Period=period,
                EvaluationPeriods=eval_periods,
                Threshold=threshold,
```

```
                ComparisonOperator=comparison_op,
            )
            logger.info(
                "Added alarm %s to track metric %s.%s.",
                alarm_name,
                metric_namespace,
                metric_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't add alarm %s to metric %s.%s",
                alarm_name,
                metric_namespace,
                metric_name,
            )
            raise
        else:
            return alarm
```

- Einzelheiten zur API finden Sie PutMetricAlarmin AWS SDK for Python (Boto3) API Reference.

Ruby

SDK für Ruby

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
# Creates or updates an alarm in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param alarm_name [String] The name of the alarm.
# @param alarm_description [String] A description about the alarm.
# @param metric_name [String] The name of the metric associated with the alarm.
```

```
# @param alarm_actions [Array] A list of Strings representing the
#   Amazon Resource Names (ARNs) to execute when the alarm transitions to the
#   ALARM state.
# @param namespace [String] The namespace for the metric to alarm on.
# @param statistic [String] The statistic for the metric.
# @param dimensions [Array] A list of dimensions for the metric, specified as
#   Aws::CloudWatch::Types::Dimension.
# @param period [Integer] The number of seconds before re-evaluating the metric.
# @param unit [String] The unit of measure for the statistic.
# @param evaluation_periods [Integer] The number of periods over which data is
#   compared to the specified threshold.
# @param theshold [Float] The value against which the specified statistic is
 compared.
# @param comparison_operator [String] The arithmetic operation to use when
#   comparing the specified statistic and threshold.
# @return [Boolean] true if the alarm was created or updated; otherwise, false.
# @example
#   exit 1 unless alarm_created_or_updated?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'ObjectsInBucket',
#     'Objects exist in this bucket for more than 1 day.',
#     'NumberOfObjects',
#     ['arn:aws:sns:us-east-1:111111111111:Default_CloudWatch_Alarms_Topic'],
#     'AWS/S3',
#     'Average',
#     [
#       {
#         name: 'BucketName',
#         value: 'doc-example-bucket'
#       },
#       {
#         name: 'StorageType',
#         value: 'AllStorageTypes'
#       }
#     ],
#     86_400,
#     'Count',
#     1,
#     1,
#     'GreaterThanThreshold'
#   )
def alarm_created_or_updated?(
  cloudwatch_client,
  alarm_name,
```

```
  alarm_description,
  metric_name,
  alarm_actions,
  namespace,
  statistic,
  dimensions,
  period,
  unit,
  evaluation_periods,
  threshold,
  comparison_operator
)
  cloudwatch_client.put_metric_alarm(
    alarm_name: alarm_name,
    alarm_description: alarm_description,
    metric_name: metric_name,
    alarm_actions: alarm_actions,
    namespace: namespace,
    statistic: statistic,
    dimensions: dimensions,
    period: period,
    unit: unit,
    evaluation_periods: evaluation_periods,
    threshold: threshold,
    comparison_operator: comparison_operator
  )
  return true
rescue StandardError => e
  puts "Error creating alarm: #{e.message}"
  return false
end
```

- Einzelheiten zur API finden Sie PutMetricAlarmin der AWS SDK for Ruby API-Referenz.

SAP ABAP

SDK für SAP ABAP

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
TRY.
    lo_cwt->putmetricalarm(
      iv_alarmname                = iv_alarm_name
      iv_comparisonoperator       = iv_comparison_operator
      iv_evaluationperiods        = iv_evaluation_periods
      iv_metricname               = iv_metric_name
      iv_namespace                = iv_namespace
      iv_statistic                = iv_statistic
      iv_threshold                = iv_threshold
      iv_actionsenabled           = iv_actions_enabled
      iv_alarmdescription         = iv_alarm_description
      iv_unit                     = iv_unit
      iv_period                   = iv_period
      it_dimensions               = it_dimensions
    ).
    MESSAGE 'Alarm created.' TYPE 'I'.
  CATCH /aws1/cx_cwtlimitexceededfault.
    MESSAGE 'The request processing has exceeded the limit' TYPE 'E'.
ENDTRY.
```

- Einzelheiten zur API finden Sie PutMetricAlarmin der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`PutMetricData`** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird`PutMetricData`.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- Erste Schritte mit CloudWatch-Metriken, -Dashboards und -Alarmen
- Metriken und Alarme verwalten

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
   /// <summary>
   /// Add some metric data using a call to a wrapper class.
   /// </summary>
   /// <param name="customMetricName">The metric name.</param>
   /// <param name="customMetricNamespace">The metric namespace.</param>
   /// <returns></returns>
   private static async Task<List<MetricDatum>> PutRandomMetricData(string
customMetricName,
       string customMetricNamespace)
   {
       List<MetricDatum> customData = new List<MetricDatum>();
       Random rnd = new Random();

       // Add 10 random values up to 100, starting with a timestamp 15 minutes
in the past.
       var utcNowMinus15 = DateTime.UtcNow.AddMinutes(-15);
       for (int i = 0; i < 10; i++)
       {
           var metricValue = rnd.Next(0, 100);
           customData.Add(
               new MetricDatum
               {
                   MetricName = customMetricName,
                   Value = metricValue,
```

```
                    TimestampUtc = utcNowMinus15.AddMinutes(i)
                }
            );
        }

        await _cloudWatchWrapper.PutMetricData(customMetricNamespace,
customData);
        return customData;
    }

    /// <summary>
    /// Wrapper to add metric data to a CloudWatch metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricData">A data object for the metric data.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutMetricData(string metricNamespace,
        List<MetricDatum> metricData)
    {
        var putDataResponse = await _amazonCloudWatch.PutMetricDataAsync(
            new PutMetricDataRequest()
            {
                MetricData = metricData,
                Namespace = metricNamespace,
            });

        return putDataResponse.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Einzelheiten zur API finden Sie PutMetricDatain der AWS SDK for .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Binden Sie die erforderlichen Dateien ein.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/PutMetricDataRequest.h>
#include <iostream>
```

Einfügen von Daten in eine Metrik

```
        Aws::CloudWatch::CloudWatchClient cw;

        Aws::CloudWatch::Model::Dimension dimension;
        dimension.SetName("UNIQUE_PAGES");
        dimension.SetValue("URLS");

        Aws::CloudWatch::Model::MetricDatum datum;
        datum.SetMetricName("PAGES_VISITED");
        datum.SetUnit(Aws::CloudWatch::Model::StandardUnit::None);
        datum.SetValue(data_point);
        datum.AddDimensions(dimension);

        Aws::CloudWatch::Model::PutMetricDataRequest request;
        request.SetNamespace("SITE/TRAFFIC");
        request.AddMetricData(datum);

        auto outcome = cw.PutMetricData(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to put sample metric data:" <<
                outcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout << "Successfully put sample metric data" << std::endl;
        }
```

- Einzelheiten zur API finden Sie PutMetricDatain der AWS SDK for C++ API-Referenz.

CLI

AWS CLI

Um eine benutzerdefinierte Metrik auf Amazon zu veröffentlichen CloudWatch

Im folgenden Beispiel wird der `put-metric-data` Befehl verwendet, um eine benutzerdefinierte Metrik auf Amazon zu veröffentlichen CloudWatch:

```
aws cloudwatch put-metric-data --namespace "Usage Metrics" --metric-data file://
metric.json
```

Die Werte für die Metrik selbst werden in der JSON-Datei `metric.json` gespeichert.

Hier ist der Inhalt dieser Datei:

```
[
  {
    "MetricName": "New Posts",
    "Timestamp": "Wednesday, June 12, 2013 8:28:20 PM",
    "Value": 0.50,
    "Unit": "Count"
  }
]
```

Weitere Informationen finden Sie unter Veröffentlichen benutzerdefinierter Metriken im Amazon CloudWatch Developer Guide.

So geben Sie mehrere Dimensionen an

Das folgende Beispiel zeigt, wie mehrere Dimensionen angegeben werden können. Jede Dimension wird als Name/Wert-Paar angegeben. Mehrere Dimensionen sind durch ein Komma getrennt:

```
aws cloudwatch put-metric-data --metric-name Buffers --namespace
 MyNameSpace --unit Bytes --value 231434333 --dimensions
 InstanceID=1-23456789,InstanceType=m1.small
```

- Einzelheiten zur API finden Sie PutMetricDatain der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
   public static void addMetricDataForAlarm(CloudWatchClient cw, String
fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();

           // Set an Instant object.
           String time =
ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT);
           Instant instant = Instant.parse(time);

           MetricDatum datum = MetricDatum.builder()
                   .metricName(customMetricName)
                   .unit(StandardUnit.NONE)
                   .value(1001.00)
                   .timestamp(instant)
                   .build();

           MetricDatum datum2 = MetricDatum.builder()
                   .metricName(customMetricName)
                   .unit(StandardUnit.NONE)
                   .value(1002.00)
                   .timestamp(instant)
                   .build();
```

```
            List<MetricDatum> metricDataList = new ArrayList<>();
            metricDataList.add(datum);
            metricDataList.add(datum2);

            PutMetricDataRequest request = PutMetricDataRequest.builder()
                    .namespace(customMetricNamespace)
                    .metricData(metricDataList)
                    .build();

            cw.putMetricData(request);
            System.out.println("Added metric values for for metric " +
 customMetricName);

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie PutMetricDatain der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Importieren Sie das SDK- und Client-Module und rufen Sie die API auf.

```
import { PutMetricDataCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  // See https://docs.aws.amazon.com/AmazonCloudWatch/latest/APIReference/
API_PutMetricData.html#API_PutMetricData_RequestParameters
  // and https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
publishingMetrics.html
```

```
  // for more information about the parameters in this command.
  const command = new PutMetricDataCommand({
    MetricData: [
      {
        MetricName: "PAGES_VISITED",
        Dimensions: [
          {
            Name: "UNIQUE_PAGES",
            Value: "URLS",
          },
        ],
        Unit: "None",
        Value: 1.0,
      },
    ],
    Namespace: "SITE/TRAFFIC",
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Erstellen Sie den Client in einem separaten Modul und exportieren Sie ihn.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Weitere Informationen finden Sie im AWS SDK for JavaScript -Entwicklerhandbuch.
- Einzelheiten zur API finden Sie PutMetricDatain der AWS SDK for JavaScript API-Referenz.

SDK für JavaScript (v2)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

// Create parameters JSON for putMetricData
var params = {
  MetricData: [
    {
      MetricName: "PAGES_VISITED",
      Dimensions: [
        {
          Name: "UNIQUE_PAGES",
          Value: "URLS",
        },
      ],
      Unit: "None",
      Value: 1.0,
    },
  ],
  Namespace: "SITE/TRAFFIC",
};

cw.putMetricData(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", JSON.stringify(data));
  }
});
```

- Weitere Informationen finden Sie im AWS SDK for JavaScript -Entwicklerhandbuch.
- Einzelheiten zur API finden Sie PutMetricDatain der AWS SDK for JavaScript API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
suspend fun addMetricDataForAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set an Instant object.
    val time =
 ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT)
    val instant = Instant.parse(time)
    val datum = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1001.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }

    val datum2 = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1002.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }

    val metricDataList = ArrayList<MetricDatum>()
    metricDataList.add(datum)
```

```
    metricDataList.add(datum2)

    val request = PutMetricDataRequest {
        namespace = customMetricNamespace
        metricData = metricDataList
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricData(request)
        println("Added metric values for for metric $customMetricName")
    }
}
```

- API-Details finden Sie PutMetricDatain der API-Referenz zum AWS SDK für Kotlin.

PowerShell

Tools für PowerShell

Beispiel 1: Erstellt ein neues MetricDatum Objekt und schreibt es in Amazon Web Services CloudWatch Metrics.

```
### Create a MetricDatum .NET object
$Metric = New-Object -TypeName Amazon.CloudWatch.Model.MetricDatum
$Metric.Timestamp = [DateTime]::UtcNow
$Metric.MetricName = 'CPU'
$Metric.Value = 50

### Write the metric data to the CloudWatch service
Write-CWMetricData -Namespace instance1 -MetricData $Metric
```

- Einzelheiten zur API finden Sie unter PutMetricData AWS Tools for PowerShellCmdlet-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def put_metric_data(self, namespace, name, value, unit):
        """
        Sends a single data value to CloudWatch for a metric. This metric is
 given
        a timestamp of the current UTC time.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param value: The value of the metric.
        :param unit: The unit of the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[{"MetricName": name, "Value": value, "Unit": unit}],
            )
            logger.info("Put data for metric %s.%s", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data for metric %s.%s", namespace,
 name)
            raise
```

Setze einen Datensatz in eine CloudWatch Metrik um.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def put_metric_data_set(self, namespace, name, timestamp, unit, data_set):
        """
        Sends a set of data to CloudWatch for a metric. All of the data in the
 set
        have the same timestamp and unit.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param timestamp: The UTC timestamp for the metric.
        :param unit: The unit of the metric.
        :param data_set: The set of data to send. This set is a dictionary that
                         contains a list of values and a list of corresponding
 counts.
                         The value and count lists must be the same length.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[
                    {
                        "MetricName": name,
                        "Timestamp": timestamp,
                        "Values": data_set["values"],
                        "Counts": data_set["counts"],
                        "Unit": unit,
                    }
                ],
```

```
            )
            logger.info("Put data set for metric %s.%s.", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data set for metric %s.%s.",
namespace, name)
            raise
```

- Einzelheiten zur API finden Sie PutMetricDatain AWS SDK for Python (Boto3) API Reference.

Ruby

SDK für Ruby

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
require "aws-sdk-cloudwatch"

# Adds a datapoint to a metric in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param metric_namespace [String] The namespace of the metric to add the
#   datapoint to.
# @param metric_name [String] The name of the metric to add the datapoint to.
# @param dimension_name [String] The name of the dimension to add the
#   datapoint to.
# @param dimension_value [String] The value of the dimension to add the
#   datapoint to.
# @param metric_value [Float] The value of the datapoint.
# @param metric_unit [String] The unit of measurement for the datapoint.
# @return [Boolean]
# @example
#   exit 1 unless datapoint_added_to_metric?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
```

```
#     'SITE/TRAFFIC',
#     'UniqueVisitors',
#     'SiteName',
#     'example.com',
#     5_885.0,
#     'Count'
#   )
def datapoint_added_to_metric?(
  cloudwatch_client,
  metric_namespace,
  metric_name,
  dimension_name,
  dimension_value,
  metric_value,
  metric_unit
)
  cloudwatch_client.put_metric_data(
    namespace: metric_namespace,
    metric_data: [
      {
        metric_name: metric_name,
        dimensions: [
          {
            name: dimension_name,
            value: dimension_value
          }
        ],
        value: metric_value,
        unit: metric_unit
      }
    ]
  )
  puts "Added data about '#{metric_name}' to namespace " \
    "'#{metric_namespace}'."
  return true
rescue StandardError => e
  puts "Error adding data about '#{metric_name}' to namespace " \
    "'#{metric_namespace}': #{e.message}"
  return false
end
```

- Einzelheiten zur API finden Sie PutMetricDatain der AWS SDK for Ruby API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Szenarien für die CloudWatch Verwendung von AWS SDKs

Die folgenden Codebeispiele zeigen Ihnen, wie Sie allgemeine Szenarien CloudWatch mit AWS SDKs implementieren. Diese Szenarien zeigen Ihnen, wie Sie bestimmte Aufgaben erledigen können, indem Sie darin mehrere Funktionen aufrufen. CloudWatch Jedes Szenario enthält einen Link zu GitHub, über den Sie Anweisungen zum Einrichten und Ausführen des Codes finden.

Beispiele

- Erste Schritte mit CloudWatch Alarmen mithilfe eines AWS SDK
- Erste Schritte mit CloudWatch Metriken, Dashboards und Alarmen mithilfe eines SDK AWS
- CloudWatch Metriken und Alarme mithilfe eines AWS SDK verwalten

Erste Schritte mit CloudWatch Alarmen mithilfe eines AWS SDK

Wie das aussehen kann, sehen Sie am nachfolgenden Beispielcode:

- Erstellen Sie einen Alarm.
- Deaktivieren Sie Alarmaktionen.
- Beschreiben Sie einen Alarm.
- Löschen Sie einen Alarm.

SAP ABAP

SDK für SAP ABAP

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
    DATA lt_alarmnames TYPE /aws1/cl_cwtalarmnames_w=>tt_alarmnames.
    DATA lo_alarmname TYPE REF TO /aws1/cl_cwtalarmnames_w.

    "Create an alarm"
    TRY.
        lo_cwt->putmetricalarm(
          iv_alarmname                = iv_alarm_name
          iv_comparisonoperator       = iv_comparison_operator
          iv_evaluationperiods        = iv_evaluation_periods
          iv_metricname               = iv_metric_name
          iv_namespace                = iv_namespace
          iv_statistic                = iv_statistic
          iv_threshold                = iv_threshold
          iv_actionsenabled           = iv_actions_enabled
          iv_alarmdescription         = iv_alarm_description
          iv_unit                     = iv_unit
          iv_period                   = iv_period
          it_dimensions               = it_dimensions
        ).
        MESSAGE 'Alarm created' TYPE 'I'.
      CATCH /aws1/cx_cwtlimitexceededfault.
        MESSAGE 'The request processing has exceeded the limit' TYPE 'E'.
    ENDTRY.

    "Create an ABAP internal table for the created alarm."
    CREATE OBJECT lo_alarmname EXPORTING iv_value = iv_alarm_name.
    INSERT lo_alarmname INTO TABLE lt_alarmnames.

    "Disable alarm actions."
    TRY.
        lo_cwt->disablealarmactions(
          it_alarmnames               = lt_alarmnames
        ).
        MESSAGE 'Alarm actions disabled' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_disablealarm_exception).
        DATA(lv_disablealarm_error) = |"{ lo_disablealarm_exception-
>av_err_code }" - { lo_disablealarm_exception->av_err_msg }|.
        MESSAGE lv_disablealarm_error TYPE 'E'.
    ENDTRY.

    "Describe alarm using the same ABAP internal table."
    TRY.
        oo_result = lo_cwt->describealarms(                     " oo_result is
 returned for testing purpose "
```

```
          it_alarmnames                = lt_alarmnames
        ).
        MESSAGE 'Alarms retrieved' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_describealarms_exception).
        DATA(lv_describealarms_error) = |"{ lo_describealarms_exception-
>av_err_code }" - { lo_describealarms_exception->av_err_msg }|.
        MESSAGE lv_describealarms_error TYPE 'E'.
    ENDTRY.

    "Delete alarm."
    TRY.
        lo_cwt->deletealarms(
          it_alarmnames = lt_alarmnames
        ).
        MESSAGE 'Alarms deleted' TYPE 'I'.
      CATCH /aws1/cx_cwtresourcenotfound .
        MESSAGE 'Resource being access is not found.' TYPE 'E'.
    ENDTRY.
```

- Weitere API-Informationen finden Sie in den folgenden Themen der API-Referenz zum AWS SDK für SAP ABAP.
  - DeleteAlarms
  - DescribeAlarms
  - DisableAlarmActions
  - PutMetricAlarm

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erste Schritte mit CloudWatch Metriken, Dashboards und Alarmen mithilfe eines SDK AWS

Die folgenden Code-Beispiele veranschaulichen Folgendes:

- Listet CloudWatch Namespaces und Metriken auf.
- Rufen Sie Statistiken für eine Metrik und die geschätzte Fakturierung ab.
- Erstellen und aktualisieren Sie ein Dashboard.

- Erstellen Sie eine Metrik und fügen Sie ihr Daten hinzu.
- Erstellen und lösen Sie einen Alarm aus und zeigen Sie dann den Alarmverlauf an.
- Fügen Sie einen Anomaliedetektor hinzu.
- Ermitteln Sie ein Metrik-Image, dann bereinigen Sie die Ressourcen.

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

Führen Sie ein interaktives Szenario an einer Eingabeaufforderung aus.

```
public class CloudWatchScenario
{
    /*
    Before running this .NET code example, set up your development environment,
 including your credentials.

    To enable billing metrics and statistics for this example, make sure billing
 alerts are enabled for your account:
    https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
monitor_estimated_charges_with_cloudwatch.html#turning_on_billing_metrics

    This .NET example performs the following tasks:
        1. List and select a CloudWatch namespace.
        2. List and select a CloudWatch metric.
        3. Get statistics for a CloudWatch metric.
        4. Get estimated billing statistics for the last week.
        5. Create a new CloudWatch dashboard with two metrics.
        6. List current CloudWatch dashboards.
        7. Create a CloudWatch custom metric and add metric data.
        8. Add the custom metric to the dashboard.
        9. Create a CloudWatch alarm for the custom metric.
       10. Describe current CloudWatch alarms.
       11. Get recent data for the custom metric.
       12. Add data to the custom metric to trigger the alarm.
```

```
      13. Wait for an alarm state.
      14. Get history for the CloudWatch alarm.
      15. Add an anomaly detector.
      16. Describe current anomaly detectors.
      17. Get and display a metric image.
      18. Clean up resources.
   */

   private static ILogger logger = null!;
   private static CloudWatchWrapper _cloudWatchWrapper = null!;
   private static IConfiguration _configuration = null!;
   private static readonly List<string> _statTypes = new List<string>
{ "SampleCount", "Average", "Sum", "Minimum", "Maximum" };
   private static SingleMetricAnomalyDetector? anomalyDetector = null!;

   static async Task Main(string[] args)
   {
       // Set up dependency injection for the Amazon service.
       using var host = Host.CreateDefaultBuilder(args)
           .ConfigureLogging(logging =>
               logging.AddFilter("System", LogLevel.Debug)
                   .AddFilter<DebugLoggerProvider>("Microsoft",
LogLevel.Information)
                   .AddFilter<ConsoleLoggerProvider>("Microsoft",
LogLevel.Trace))
           .ConfigureServices((_, services) =>
           services.AddAWSService<IAmazonCloudWatch>()
           .AddTransient<CloudWatchWrapper>()
       )
       .Build();

       _configuration = new ConfigurationBuilder()
           .SetBasePath(Directory.GetCurrentDirectory())
           .AddJsonFile("settings.json") // Load settings from .json file.
           .AddJsonFile("settings.local.json",
               true) // Optionally, load local settings.
           .Build();

       logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
           .CreateLogger<CloudWatchScenario>();

       _cloudWatchWrapper =
host.Services.GetRequiredService<CloudWatchWrapper>();
```

```
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Welcome to the Amazon CloudWatch example scenario.");
        Console.WriteLine(new string('-', 80));

        try
        {
            var selectedNamespace = await SelectNamespace();
            var selectedMetric = await SelectMetric(selectedNamespace);
            await GetAndDisplayMetricStatistics(selectedNamespace,
selectedMetric);
            await GetAndDisplayEstimatedBilling();
            await CreateDashboardWithMetrics();
            await ListDashboards();
            await CreateNewCustomMetric();
            await AddMetricToDashboard();
            await CreateMetricAlarm();
            await DescribeAlarms();
            await GetCustomMetricData();
            await AddMetricDataForAlarm();
            await CheckForMetricAlarm();
            await GetAlarmHistory();
            anomalyDetector = await AddAnomalyDetector();
            await DescribeAnomalyDetectors();
            await GetAndOpenMetricImage();
            await CleanupResources();
        }
        catch (Exception ex)
        {
            logger.LogError(ex, "There was a problem executing the scenario.");
            await CleanupResources();
        }

    }

    /// <summary>
    /// Select a namespace.
    /// </summary>
    /// <returns>The selected namespace.</returns>
    private static async Task<string> SelectNamespace()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"1. Select a CloudWatch Namespace from a list of
Namespaces.");
        var metrics = await _cloudWatchWrapper.ListMetrics();
```

```
        // Get a distinct list of namespaces.
        var namespaces = metrics.Select(m => m.Namespace).Distinct().ToList();
        for (int i = 0; i < namespaces.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {namespaces[i]}");
        }

        var namespaceChoiceNumber = 0;
        while (namespaceChoiceNumber < 1 || namespaceChoiceNumber >
namespaces.Count)
        {
            Console.WriteLine(
                "Select a namespace by entering a number from the preceding
list:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out namespaceChoiceNumber);
        }

        var selectedNamespace = namespaces[namespaceChoiceNumber - 1];

        Console.WriteLine(new string('-', 80));

        return selectedNamespace;
    }

    /// <summary>
    /// Select a metric from a namespace.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <returns>The metric name.</returns>
    private static async Task<Metric> SelectMetric(string metricNamespace)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"2. Select a CloudWatch metric from a namespace.");

        var namespaceMetrics = await
_cloudWatchWrapper.ListMetrics(metricNamespace);

        for (int i = 0; i < namespaceMetrics.Count && i < 15; i++)
        {
            var dimensionsWithValues = namespaceMetrics[i].Dimensions
                .Where(d => !string.Equals("None", d.Value));
            Console.WriteLine($"\t{i + 1}. {namespaceMetrics[i].MetricName} " +
```

```
                                $"{string.Join(", :", dimensionsWithValues.Select(d
=> d.Value))}");
        }

        var metricChoiceNumber = 0;
        while (metricChoiceNumber < 1 || metricChoiceNumber >
namespaceMetrics.Count)
        {
            Console.WriteLine(
                "Select a metric by entering a number from the preceding list:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out metricChoiceNumber);
        }

        var selectedMetric = namespaceMetrics[metricChoiceNumber - 1];

        Console.WriteLine(new string('-', 80));

        return selectedMetric;
    }

    /// <summary>
    /// Get and display metric statistics for a specific metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task GetAndDisplayMetricStatistics(string
metricNamespace, Metric metric)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"3. Get CloudWatch metric statistics for the last
day.");

        for (int i = 0; i < _statTypes.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {_statTypes[i]}");
        }

        var statisticChoiceNumber = 0;
        while (statisticChoiceNumber < 1 || statisticChoiceNumber >
_statTypes.Count)
        {
            Console.WriteLine(
```

```
                "Select a metric statistic by entering a number from the
preceding list:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out statisticChoiceNumber);
        }

        var selectedStatistic = _statTypes[statisticChoiceNumber - 1];
        var statisticsList = new List<string> { selectedStatistic };

        var metricStatistics = await
_cloudWatchWrapper.GetMetricStatistics(metricNamespace, metric.MetricName,
statisticsList, metric.Dimensions, 1, 60);

        if (!metricStatistics.Any())
        {
            Console.WriteLine($"No {selectedStatistic} statistics found for
{metric} in namespace {metricNamespace}.");
        }

        metricStatistics = metricStatistics.OrderBy(s => s.Timestamp).ToList();
        for (int i = 0; i < metricStatistics.Count && i < 10; i++)
        {
            var metricStat = metricStatistics[i];
            var statValue =
metricStat.GetType().GetProperty(selectedStatistic)!.GetValue(metricStat, null);
            Console.WriteLine($"\t{i + 1}. Timestamp
{metricStatistics[i].Timestamp:G} {selectedStatistic}: {statValue}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get and display estimated billing statistics.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task GetAndDisplayEstimatedBilling()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"4. Get CloudWatch estimated billing for the last
week.");
```

```
        var billingStatistics = await SetupBillingStatistics();

        for (int i = 0; i < billingStatistics.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. Timestamp
{billingStatistics[i].Timestamp:G} : {billingStatistics[i].Maximum}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get billing statistics using a call to a wrapper class.
    /// </summary>
    /// <returns>A collection of billing statistics.</returns>
    private static async Task<List<Datapoint>> SetupBillingStatistics()
    {
        // Make a request for EstimatedCharges with a period of one day for the
past seven days.
        var billingStatistics = await _cloudWatchWrapper.GetMetricStatistics(
            "AWS/Billing",
            "EstimatedCharges",
            new List<string>() { "Maximum" },
            new List<Dimension>() { new Dimension { Name = "Currency", Value =
"USD" } },
            7,
            86400);

        billingStatistics = billingStatistics.OrderBy(n => n.Timestamp).ToList();

        return billingStatistics;
    }

    /// <summary>
    /// Create a dashboard with metrics.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task CreateDashboardWithMetrics()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"5. Create a new CloudWatch dashboard with metrics.");
        var dashboardName = _configuration["dashboardName"];
```

```
        var newDashboard = new DashboardModel();
        _configuration.GetSection("dashboardExampleBody").Bind(newDashboard);
        var newDashboardString = JsonSerializer.Serialize(
            newDashboard,
            new JsonSerializerOptions
            {
                DefaultIgnoreCondition = JsonIgnoreCondition.WhenWritingNull
            });
        var validationMessages =
            await _cloudWatchWrapper.PutDashboard(dashboardName,
newDashboardString);

        Console.WriteLine(validationMessages.Any() ? $"\tValidation messages:" :
null);
        for (int i = 0; i < validationMessages.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {validationMessages[i].Message}");
        }
        Console.WriteLine($"\tDashboard {dashboardName} was created.");
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// List dashboards.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task ListDashboards()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"6. List the CloudWatch dashboards in the current
account.");

        var dashboards = await _cloudWatchWrapper.ListDashboards();

        for (int i = 0; i < dashboards.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {dashboards[i].DashboardName}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Create and add data for a new custom metric.
```

```
   /// </summary>
   /// <returns>Async task.</returns>
   private static async Task CreateNewCustomMetric()
   {
       Console.WriteLine(new string('-', 80));
       Console.WriteLine($"7. Create and add data for a new custom metric.");

       var customMetricNamespace = _configuration["customMetricNamespace"];
       var customMetricName = _configuration["customMetricName"];

       var customData = await PutRandomMetricData(customMetricName,
customMetricNamespace);

       var valuesString = string.Join(',', customData.Select(d => d.Value));
       Console.WriteLine($"\tAdded metric values for for metric
{customMetricName}: \n\t{valuesString}");

       Console.WriteLine(new string('-', 80));
   }


   /// <summary>
   /// Add some metric data using a call to a wrapper class.
   /// </summary>
   /// <param name="customMetricName">The metric name.</param>
   /// <param name="customMetricNamespace">The metric namespace.</param>
   /// <returns></returns>
   private static async Task<List<MetricDatum>> PutRandomMetricData(string
customMetricName,
       string customMetricNamespace)
   {
       List<MetricDatum> customData = new List<MetricDatum>();
       Random rnd = new Random();

       // Add 10 random values up to 100, starting with a timestamp 15 minutes
in the past.
       var utcNowMinus15 = DateTime.UtcNow.AddMinutes(-15);
       for (int i = 0; i < 10; i++)
       {
           var metricValue = rnd.Next(0, 100);
           customData.Add(
               new MetricDatum
               {
                   MetricName = customMetricName,
```

```
                    Value = metricValue,
                    TimestampUtc = utcNowMinus15.AddMinutes(i)
                }
            );
        }

        await _cloudWatchWrapper.PutMetricData(customMetricNamespace,
customData);
        return customData;
    }

    /// <summary>
    /// Add the custom metric to the dashboard.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task AddMetricToDashboard()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"8. Add the new custom metric to the dashboard.");

        var dashboardName = _configuration["dashboardName"];

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var validationMessages = await SetupDashboard(customMetricNamespace,
customMetricName, dashboardName);

        Console.WriteLine(validationMessages.Any() ? $"\tValidation messages:" :
null);
        for (int i = 0; i < validationMessages.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {validationMessages[i].Message}");
        }
        Console.WriteLine($"\tDashboard {dashboardName} updated with metric
{customMetricName}.");
        Console.WriteLine(new string('-', 80));
    }


    /// <summary>
    /// Set up a dashboard using a call to the wrapper class.
    /// </summary>
    /// <param name="customMetricNamespace">The metric namespace.</param>
```

```
    /// <param name="customMetricName">The metric name.</param>
    /// <param name="dashboardName">The name of the dashboard.</param>
    /// <returns>A list of validation messages.</returns>
    private static async Task<List<DashboardValidationMessage>> SetupDashboard(
        string customMetricNamespace, string customMetricName, string
dashboardName)
    {
        // Get the dashboard model from configuration.
        var newDashboard = new DashboardModel();
        _configuration.GetSection("dashboardExampleBody").Bind(newDashboard);

        // Add a new metric to the dashboard.
        newDashboard.Widgets.Add(new Widget
        {
            Height = 8,
            Width = 8,
            Y = 8,
            X = 0,
            Type = "metric",
            Properties = new Properties
            {
                Metrics = new List<List<object>>
                    { new() { customMetricNamespace, customMetricName } },
                View = "timeSeries",
                Region = "us-east-1",
                Stat = "Sum",
                Period = 86400,
                YAxis = new YAxis { Left = new Left { Min = 0, Max = 100 } },
                Title = "Custom Metric Widget",
                LiveData = true,
                Sparkline = true,
                Trend = true,
                Stacked = false,
                SetPeriodToTimeRange = false
            }
        });

        var newDashboardString = JsonSerializer.Serialize(newDashboard,
            new JsonSerializerOptions
            { DefaultIgnoreCondition = JsonIgnoreCondition.WhenWritingNull });
        var validationMessages =
            await _cloudWatchWrapper.PutDashboard(dashboardName,
newDashboardString);
```

```
        return validationMessages;
    }

    /// <summary>
    /// Create a CloudWatch alarm for the new metric.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task CreateMetricAlarm()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"9. Create a CloudWatch alarm for the new metric.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var alarmName = _configuration["exampleAlarmName"];
        var accountId = _configuration["accountId"];
        var region = _configuration["region"];
        var emailTopic = _configuration["emailTopic"];
        var alarmActions = new List<string>();

        if (GetYesNoResponse(
                $"\tAdd an email action for topic {emailTopic} to alarm
{alarmName}? (y/n)"))
        {
            _cloudWatchWrapper.AddEmailAlarmAction(accountId, region, emailTopic,
alarmActions);
        }

        await _cloudWatchWrapper.PutMetricEmailAlarm(
            "Example metric alarm",
            alarmName,
            ComparisonOperator.GreaterThanOrEqualToThreshold,
            customMetricName,
            customMetricNamespace,
            100,
            alarmActions);

        Console.WriteLine($"\tAlarm {alarmName} added for metric
{customMetricName}.");
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
```

```
    /// Describe Alarms.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task DescribeAlarms()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"10. Describe CloudWatch alarms in the current
account.");

        var alarms = await _cloudWatchWrapper.DescribeAlarms();
        alarms = alarms.OrderByDescending(a => a.StateUpdatedTimestamp).ToList();

        for (int i = 0; i < alarms.Count && i < 10; i++)
        {
            var alarm = alarms[i];
            Console.WriteLine($"\t{i + 1}. {alarm.AlarmName}");
            Console.WriteLine($"\tState: {alarm.StateValue} for
{alarm.MetricName} {alarm.ComparisonOperator} {alarm.Threshold}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get the recent data for the metric.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task GetCustomMetricData()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"11. Get current data for new custom metric.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
        var accountId = _configuration["accountId"];

        var query = new List<MetricDataQuery>
        {
            new MetricDataQuery
            {
                AccountId = accountId,
                Id = "m1",
                Label = "Custom Metric Data",
                MetricStat = new MetricStat
```

```
                {
                    Metric = new Metric
                    {
                        MetricName = customMetricName,
                        Namespace = customMetricNamespace,
                    },
                    Period = 1,
                    Stat = "Maximum"
                }
            }
        };

        var metricData = await _cloudWatchWrapper.GetMetricData(
            20,
            true,
            DateTime.UtcNow.AddMinutes(1),
            20,
            query);

        for (int i = 0; i < metricData.Count; i++)
        {
            for (int j = 0; j < metricData[i].Values.Count; j++)
            {
                Console.WriteLine(
                    $"\tTimestamp {metricData[i].Timestamps[j]:G} Value:
{metricData[i].Values[j]}");
            }
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Add metric data to trigger an alarm.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task AddMetricDataForAlarm()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"12. Add metric data to the custom metric to trigger
an alarm.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
```

```
        var nowUtc = DateTime.UtcNow;
        List<MetricDatum> customData = new List<MetricDatum>
        {
            new MetricDatum
            {
                MetricName = customMetricName,
                Value = 101,
                TimestampUtc = nowUtc.AddMinutes(-2)
            },
            new MetricDatum
            {
                MetricName = customMetricName,
                Value = 101,
                TimestampUtc = nowUtc.AddMinutes(-1)
            },
            new MetricDatum
            {
                MetricName = customMetricName,
                Value = 101,
                TimestampUtc = nowUtc
            }
        };
        var valuesString = string.Join(',', customData.Select(d => d.Value));
        Console.WriteLine($"\tAdded metric values for for metric
{customMetricName}: \n\t{valuesString}");
        await _cloudWatchWrapper.PutMetricData(customMetricNamespace,
customData);

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Check for a metric alarm using the DescribeAlarmsForMetric action.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task CheckForMetricAlarm()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"13. Checking for an alarm state.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
        var hasAlarm = false;
        var retries = 10;
```

```
        while (!hasAlarm && retries > 0)
        {
            var alarms = await
_cloudWatchWrapper.DescribeAlarmsForMetric(customMetricNamespace,
customMetricName);
            hasAlarm = alarms.Any(a => a.StateValue == StateValue.ALARM);
            retries--;
            Thread.Sleep(20000);
        }

        Console.WriteLine(hasAlarm
            ? $"\tAlarm state found for {customMetricName}."
            : $"\tNo Alarm state found for {customMetricName} after 10
retries.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get history for an alarm.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task GetAlarmHistory()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"14. Get alarm history.");

        var exampleAlarmName = _configuration["exampleAlarmName"];

        var alarmHistory = await
_cloudWatchWrapper.DescribeAlarmHistory(exampleAlarmName, 2);

        for (int i = 0; i < alarmHistory.Count; i++)
        {
            var history = alarmHistory[i];
            Console.WriteLine($"\t{i + 1}. {history.HistorySummary}, time
{history.Timestamp:g}");
        }
        if (!alarmHistory.Any())
        {
            Console.WriteLine($"\tNo alarm history data found for
{exampleAlarmName}.");
        }
```

```
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Add an anomaly detector.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task<SingleMetricAnomalyDetector> AddAnomalyDetector()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"15. Add an anomaly detector.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var detector = new SingleMetricAnomalyDetector
        {
            MetricName = customMetricName,
            Namespace = customMetricNamespace,
            Stat = "Maximum"
        };
        await _cloudWatchWrapper.PutAnomalyDetector(detector);
        Console.WriteLine($"\tAdded anomaly detector for metric
{customMetricName}.");

        Console.WriteLine(new string('-', 80));
        return detector;
    }

    /// <summary>
    /// Describe anomaly detectors.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task DescribeAnomalyDetectors()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"16. Describe anomaly detectors in the current
account.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
```

```
        var detectors = await
_cloudWatchWrapper.DescribeAnomalyDetectors(customMetricNamespace,
customMetricName);

        for (int i = 0; i < detectors.Count; i++)
        {
            var detector = detectors[i];
            Console.WriteLine($"\t{i + 1}.
{detector.SingleMetricAnomalyDetector.MetricName}, state
{detector.StateValue}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Fetch and open a metrics image for a CloudWatch metric and namespace.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task GetAndOpenMetricImage()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("17. Get a metric image from CloudWatch.");

        Console.WriteLine($"\tGetting Image data for custom metric.");
        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var memoryStream = await
_cloudWatchWrapper.GetTimeSeriesMetricImage(customMetricNamespace,
customMetricName, "Maximum", 10);
        var file = _cloudWatchWrapper.SaveMetricImage(memoryStream,
"MetricImages");

        ProcessStartInfo info = new ProcessStartInfo();

        Console.WriteLine($"\tFile saved as {Path.GetFileName(file)}.");
        Console.WriteLine($"\tPress enter to open the image.");
        Console.ReadLine();
        info.FileName = Path.Combine("ms-photos://", file);
        info.UseShellExecute = true;
        info.CreateNoWindow = true;
        info.Verb = string.Empty;
```

```
        Process.Start(info);

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Clean up created resources.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task CleanupResources()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"18. Clean up resources.");

        var dashboardName = _configuration["dashboardName"];
        if (GetYesNoResponse($"\tDelete dashboard {dashboardName}? (y/n)"))
        {
            Console.WriteLine($"\tDeleting dashboard.");
            var dashboardList = new List<string> { dashboardName };
            await _cloudWatchWrapper.DeleteDashboards(dashboardList);
        }

        var alarmName = _configuration["exampleAlarmName"];
        if (GetYesNoResponse($"\tDelete alarm {alarmName}? (y/n)"))
        {
            Console.WriteLine($"\tCleaning up alarms.");
            var alarms = new List<string> { alarmName };
            await _cloudWatchWrapper.DeleteAlarms(alarms);
        }

        if (GetYesNoResponse($"\tDelete anomaly detector? (y/n)") &&
anomalyDetector != null)
        {
            Console.WriteLine($"\tCleaning up anomaly detector.");

            await _cloudWatchWrapper.DeleteAnomalyDetector(
                anomalyDetector);
        }

        Console.WriteLine(new string('-', 80));
    }
```

```
    /// <summary>
    /// Get a yes or no response from the user.
    /// </summary>
    /// <param name="question">The question string to print on the console.</
param>
    /// <returns>True if the user responds with a yes.</returns>
    private static bool GetYesNoResponse(string question)
    {
        Console.WriteLine(question);
        var ynResponse = Console.ReadLine();
        var response = ynResponse != null &&
                       ynResponse.Equals("y",
                           StringComparison.InvariantCultureIgnoreCase);
        return response;
    }
}
```

Wrapper-Methoden, die vom Szenario für CloudWatch Aktionen verwendet werden.

```
/// <summary>
/// Wrapper class for Amazon CloudWatch methods.
/// </summary>
public class CloudWatchWrapper
{
    private readonly IAmazonCloudWatch _amazonCloudWatch;
    private readonly ILogger<CloudWatchWrapper> _logger;

    /// <summary>
    /// Constructor for the CloudWatch wrapper.
    /// </summary>
    /// <param name="amazonCloudWatch">The injected CloudWatch client.</param>
    /// <param name="logger">The injected logger for the wrapper.</param>
    public CloudWatchWrapper(IAmazonCloudWatch amazonCloudWatch,
 ILogger<CloudWatchWrapper> logger)

    {
        _logger = logger;
        _amazonCloudWatch = amazonCloudWatch;
    }

    /// <summary>
    /// List metrics available, optionally within a namespace.
```

```
    /// </summary>
    /// <param name="metricNamespace">Optional CloudWatch namespace to use when
listing metrics.</param>
    /// <param name="filter">Optional dimension filter.</param>
    /// <param name="metricName">Optional metric name filter.</param>
    /// <returns>The list of metrics.</returns>
    public async Task<List<Metric>> ListMetrics(string? metricNamespace = null,
DimensionFilter? filter = null, string? metricName = null)
    {
        var results = new List<Metric>();
        var paginateMetrics = _amazonCloudWatch.Paginators.ListMetrics(
            new ListMetricsRequest
            {
                Namespace = metricNamespace,
                Dimensions = filter != null ? new List<DimensionFilter>
{ filter } : null,
                MetricName = metricName
            });
        // Get the entire list using the paginator.
        await foreach (var metric in paginateMetrics.Metrics)
        {
            results.Add(metric);
        }

        return results;
    }

    /// <summary>
    /// Wrapper to get statistics for a specific CloudWatch metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <param name="statistics">The list of statistics to include.</param>
    /// <param name="dimensions">The list of dimensions to include.</param>
    /// <param name="days">The number of days in the past to include.</param>
    /// <param name="period">The period for the data.</param>
    /// <returns>A list of DataPoint objects for the statistics.</returns>
    public async Task<List<Datapoint>> GetMetricStatistics(string
metricNamespace,
        string metricName, List<string> statistics, List<Dimension> dimensions,
int days, int period)
    {
        var metricStatistics = await _amazonCloudWatch.GetMetricStatisticsAsync(
            new GetMetricStatisticsRequest()
```

```
            {
                Namespace = metricNamespace,
                MetricName = metricName,
                Dimensions = dimensions,
                Statistics = statistics,
                StartTimeUtc = DateTime.UtcNow.AddDays(-days),
                EndTimeUtc = DateTime.UtcNow,
                Period = period
            });

        return metricStatistics.Datapoints;
    }

    /// <summary>
    /// Wrapper to create or add to a dashboard with metrics.
    /// </summary>
    /// <param name="dashboardName">The name for the dashboard.</param>
    /// <param name="dashboardBody">The metric data in JSON for the dashboard.</
param>
    /// <returns>A list of validation messages for the dashboard.</returns>
    public async Task<List<DashboardValidationMessage>> PutDashboard(string
 dashboardName,
        string dashboardBody)
    {
        // Updating a dashboard replaces all contents.
        // Best practice is to include a text widget indicating this dashboard
 was created programmatically.
        var dashboardResponse = await _amazonCloudWatch.PutDashboardAsync(
            new PutDashboardRequest()
            {
                DashboardName = dashboardName,
                DashboardBody = dashboardBody
            });

        return dashboardResponse.DashboardValidationMessages;
    }


    /// <summary>
    /// Get information on a dashboard.
    /// </summary>
    /// <param name="dashboardName">The name of the dashboard.</param>
    /// <returns>A JSON object with dashboard information.</returns>
    public async Task<string> GetDashboard(string dashboardName)
```

```
    {
        var dashboardResponse = await _amazonCloudWatch.GetDashboardAsync(
            new GetDashboardRequest()
            {
                DashboardName = dashboardName
            });

        return dashboardResponse.DashboardBody;
    }


    /// <summary>
    /// Get a list of dashboards.
    /// </summary>
    /// <returns>A list of DashboardEntry objects.</returns>
    public async Task<List<DashboardEntry>> ListDashboards()
    {
        var results = new List<DashboardEntry>();
        var paginateDashboards = _amazonCloudWatch.Paginators.ListDashboards(
            new ListDashboardsRequest());
        // Get the entire list using the paginator.
        await foreach (var data in paginateDashboards.DashboardEntries)
        {
            results.Add(data);
        }

        return results;
    }

    /// <summary>
    /// Wrapper to add metric data to a CloudWatch metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricData">A data object for the metric data.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutMetricData(string metricNamespace,
        List<MetricDatum> metricData)
    {
        var putDataResponse = await _amazonCloudWatch.PutMetricDataAsync(
            new PutMetricDataRequest()
            {
                MetricData = metricData,
                Namespace = metricNamespace,
            });
```

```
        return putDataResponse.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Get an image for a metric graphed over time.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metric">The name of the metric.</param>
    /// <param name="stat">The name of the stat to chart.</param>
    /// <param name="period">The period to use for the chart.</param>
    /// <returns>A memory stream for the chart image.</returns>
    public async Task<MemoryStream> GetTimeSeriesMetricImage(string
metricNamespace, string metric, string stat, int period)
    {
        var metricImageWidget = new
        {
            title = "Example Metric Graph",
            view = "timeSeries",
            stacked = false,
            period = period,
            width = 1400,
            height = 600,
            metrics = new List<List<object>>
                { new() { metricNamespace, metric, new { stat } } }
        };

        var metricImageWidgetString =
JsonSerializer.Serialize(metricImageWidget);
        var imageResponse = await _amazonCloudWatch.GetMetricWidgetImageAsync(
            new GetMetricWidgetImageRequest()
            {
                MetricWidget = metricImageWidgetString
            });

        return imageResponse.MetricWidgetImage;
    }

    /// <summary>
    /// Save a metric image to a file.
    /// </summary>
    /// <param name="memoryStream">The MemoryStream for the metric image.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The path to the file.</returns>
```

```
    public string SaveMetricImage(MemoryStream memoryStream, string metricName)
    {
        var metricFileName = $"{metricName}_{DateTime.Now.Ticks}.png";
        using var sr = new StreamReader(memoryStream);
        // Writes the memory stream to a file.
        File.WriteAllBytes(metricFileName, memoryStream.ToArray());
        var filePath = Path.Join(AppDomain.CurrentDomain.BaseDirectory,
            metricFileName);
        return filePath;
    }

    /// <summary>
    /// Get data for CloudWatch metrics.
    /// </summary>
    /// <param name="minutesOfData">The number of minutes of data to include.</
param>
    /// <param name="useDescendingTime">True to return the data descending by
 time.</param>
    /// <param name="endDateUtc">The end date for the data, in UTC.</param>
    /// <param name="maxDataPoints">The maximum data points to include.</param>
    /// <param name="dataQueries">Optional data queries to include.</param>
    /// <returns>A list of the requested metric data.</returns>
    public async Task<List<MetricDataResult>> GetMetricData(int minutesOfData,
 bool useDescendingTime, DateTime? endDateUtc = null,
        int maxDataPoints = 0, List<MetricDataQuery>? dataQueries = null)
    {
        var metricData = new List<MetricDataResult>();
        // If no end time is provided, use the current time for the end time.
        endDateUtc ??= DateTime.UtcNow;
        var timeZoneOffset =
 TimeZoneInfo.Local.GetUtcOffset(endDateUtc.Value.ToLocalTime());
        var startTimeUtc = endDateUtc.Value.AddMinutes(-minutesOfData);
        // The timezone string should be in the format +0000, so use the timezone
 offset to format it correctly.
        var timeZoneString = $"{timeZoneOffset.Hours:D2}
{timeZoneOffset.Minutes:D2}";
        var paginatedMetricData = _amazonCloudWatch.Paginators.GetMetricData(
            new GetMetricDataRequest()
            {
                StartTimeUtc = startTimeUtc,
                EndTimeUtc = endDateUtc.Value,
                LabelOptions = new LabelOptions { Timezone = timeZoneString },
                ScanBy = useDescendingTime ? ScanBy.TimestampDescending :
 ScanBy.TimestampAscending,
```

```
                MaxDatapoints = maxDataPoints,
                MetricDataQueries = dataQueries,
            });

        await foreach (var data in paginatedMetricData.MetricDataResults)
        {
            metricData.Add(data);
        }
        return metricData;
    }

    /// <summary>
    /// Add a metric alarm to send an email when the metric passes a threshold.
    /// </summary>
    /// <param name="alarmDescription">A description of the alarm.</param>
    /// <param name="alarmName">The name for the alarm.</param>
    /// <param name="comparison">The type of comparison to use.</param>
    /// <param name="metricName">The name of the metric for the alarm.</param>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="threshold">The threshold value for the alarm.</param>
    /// <param name="alarmActions">Optional actions to execute when in an alarm
state.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutMetricEmailAlarm(string alarmDescription, string
alarmName, ComparisonOperator comparison,
        string metricName, string metricNamespace, double threshold, List<string>
alarmActions = null!)
    {
        try
        {
            var putEmailAlarmResponse = await
_amazonCloudWatch.PutMetricAlarmAsync(
                new PutMetricAlarmRequest()
                {
                    AlarmActions = alarmActions,
                    AlarmDescription = alarmDescription,
                    AlarmName = alarmName,
                    ComparisonOperator = comparison,
                    Threshold = threshold,
                    Namespace = metricNamespace,
                    MetricName = metricName,
                    EvaluationPeriods = 1,
                    Period = 10,
                    Statistic = new Statistic("Maximum"),
```

```
                    DatapointsToAlarm = 1,
                    TreatMissingData = "ignore"
                });
            return putEmailAlarmResponse.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (LimitExceededException lex)
        {
            _logger.LogError(lex, $"Unable to add alarm {alarmName}. Alarm quota
 has already been reached.");
        }

        return false;
    }

    /// <summary>
    /// Add specific email actions to a list of action strings for a CloudWatch
 alarm.
    /// </summary>
    /// <param name="accountId">The AccountId for the alarm.</param>
    /// <param name="region">The region for the alarm.</param>
    /// <param name="emailTopicName">An Amazon Simple Notification Service (SNS)
 topic for the alarm email.</param>
    /// <param name="alarmActions">Optional list of existing alarm actions to
 append to.</param>
    /// <returns>A list of string actions for an alarm.</returns>
    public List<string> AddEmailAlarmAction(string accountId, string region,
        string emailTopicName, List<string>? alarmActions = null)
    {
        alarmActions ??= new List<string>();
        var snsAlarmAction = $"arn:aws:sns:{region}:{accountId}:
{emailTopicName}";
        alarmActions.Add(snsAlarmAction);
        return alarmActions;
    }

    /// <summary>
    /// Describe the current alarms, optionally filtered by state.
    /// </summary>
    /// <param name="stateValue">Optional filter for alarm state.</param>
    /// <returns>The list of alarm data.</returns>
    public async Task<List<MetricAlarm>> DescribeAlarms(StateValue? stateValue =
 null)
    {
        List<MetricAlarm> alarms = new List<MetricAlarm>();
```

```
        var paginatedDescribeAlarms =
_amazonCloudWatch.Paginators.DescribeAlarms(
            new DescribeAlarmsRequest()
            {
                StateValue = stateValue
            });

        await foreach (var data in paginatedDescribeAlarms.MetricAlarms)
        {
            alarms.Add(data);
        }
        return alarms;
    }

    /// <summary>
    /// Describe the current alarms for a specific metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The list of alarm data.</returns>
    public async Task<List<MetricAlarm>> DescribeAlarmsForMetric(string
metricNamespace, string metricName)
    {
        var alarmsResult = await _amazonCloudWatch.DescribeAlarmsForMetricAsync(
            new DescribeAlarmsForMetricRequest()
            {
                Namespace = metricNamespace,
                MetricName = metricName
            });

        return alarmsResult.MetricAlarms;
    }

    /// <summary>
    /// Describe the history of an alarm for a number of days in the past.
    /// </summary>
    /// <param name="alarmName">The name of the alarm.</param>
    /// <param name="historyDays">The number of days in the past.</param>
    /// <returns>The list of alarm history data.</returns>
    public async Task<List<AlarmHistoryItem>> DescribeAlarmHistory(string
alarmName, int historyDays)
    {
        List<AlarmHistoryItem> alarmHistory = new List<AlarmHistoryItem>();
```

```
        var paginatedAlarmHistory =
_amazonCloudWatch.Paginators.DescribeAlarmHistory(
            new DescribeAlarmHistoryRequest()
            {
                AlarmName = alarmName,
                EndDateUtc = DateTime.UtcNow,
                HistoryItemType = HistoryItemType.StateUpdate,
                StartDateUtc = DateTime.UtcNow.AddDays(-historyDays)
            });

        await foreach (var data in paginatedAlarmHistory.AlarmHistoryItems)
        {
            alarmHistory.Add(data);
        }
        return alarmHistory;
    }

    /// <summary>
    /// Delete a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteAlarms(List<string> alarmNames)
    {
        var deleteAlarmsResult = await _amazonCloudWatch.DeleteAlarmsAsync(
            new DeleteAlarmsRequest()
            {
                AlarmNames = alarmNames
            });

        return deleteAlarmsResult.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Disable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DisableAlarmActions(List<string> alarmNames)
    {
        var disableAlarmActionsResult = await
_amazonCloudWatch.DisableAlarmActionsAsync(
            new DisableAlarmActionsRequest()
            {
```

```
                AlarmNames = alarmNames
            });

        return disableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Enable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> EnableAlarmActions(List<string> alarmNames)
    {
        var enableAlarmActionsResult = await
_amazonCloudWatch.EnableAlarmActionsAsync(
            new EnableAlarmActionsRequest()
            {
                AlarmNames = alarmNames
            });

        return enableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Add an anomaly detector for a single metric.
    /// </summary>
    /// <param name="anomalyDetector">A single metric anomaly detector.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutAnomalyDetector(SingleMetricAnomalyDetector
anomalyDetector)
    {
        var putAlarmDetectorResult = await
_amazonCloudWatch.PutAnomalyDetectorAsync(
            new PutAnomalyDetectorRequest()
            {
                SingleMetricAnomalyDetector = anomalyDetector
            });

        return putAlarmDetectorResult.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Describe anomaly detectors for a metric and namespace.
    /// </summary>
```

```
   /// <param name="metricNamespace">The namespace of the metric.</param>
   /// <param name="metricName">The metric of the anomaly detectors.</param>
   /// <returns>The list of detectors.</returns>
   public async Task<List<AnomalyDetector>> DescribeAnomalyDetectors(string
metricNamespace, string metricName)
   {
       List<AnomalyDetector> detectors = new List<AnomalyDetector>();
       var paginatedDescribeAnomalyDetectors =
_amazonCloudWatch.Paginators.DescribeAnomalyDetectors(
           new DescribeAnomalyDetectorsRequest()
           {
               MetricName = metricName,
               Namespace = metricNamespace
           });

       await foreach (var data in
paginatedDescribeAnomalyDetectors.AnomalyDetectors)
       {
           detectors.Add(data);
       }

       return detectors;
   }

   /// <summary>
   /// Delete a single metric anomaly detector.
   /// </summary>
   /// <param name="anomalyDetector">The anomaly detector to delete.</param>
   /// <returns>True if successful.</returns>
   public async Task<bool> DeleteAnomalyDetector(SingleMetricAnomalyDetector
anomalyDetector)
   {
       var deleteAnomalyDetectorResponse = await
_amazonCloudWatch.DeleteAnomalyDetectorAsync(
           new DeleteAnomalyDetectorRequest()
           {
               SingleMetricAnomalyDetector = anomalyDetector
           });

       return deleteAnomalyDetectorResponse.HttpStatusCode == HttpStatusCode.OK;
   }

   /// <summary>
   /// Delete a list of CloudWatch dashboards.
```

```
    /// </summary>
    /// <param name="dashboardNames">List of dashboard names to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteDashboards(List<string> dashboardNames)
    {
        var deleteDashboardsResponse = await
 _amazonCloudWatch.DeleteDashboardsAsync(
            new DeleteDashboardsRequest()
            {
                DashboardNames = dashboardNames
            });

        return deleteDashboardsResponse.HttpStatusCode == HttpStatusCode.OK;
    }
}
```

- API-Details finden Sie in den folgenden Themen der AWS SDK for .NET -API-Referenz.
  - DeleteAlarms
  - DeleteAnomalyDetector
  - DeleteDashboards
  - DescribeAlarmHistory
  - DescribeAlarms
  - DescribeAlarmsForMetric
  - DescribeAnomalyDetectors
  - GetMetricData
  - GetMetricStatistics
  - GetMetricWidgetImage
  - ListMetrics
  - PutAnomalyDetector
  - PutDashboard
  - PutMetricAlarm
  - PutMetricData

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
import com.fasterxml.jackson.core.JsonFactory;
import com.fasterxml.jackson.core.JsonParser;
import com.fasterxml.jackson.databind.ObjectMapper;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.core.SdkBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.AlarmHistoryItem;
import software.amazon.awssdk.services.cloudwatch.model.AlarmType;
import software.amazon.awssdk.services.cloudwatch.model.AnomalyDetector;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.ComparisonOperator;
import
 software.amazon.awssdk.services.cloudwatch.model.DashboardValidationMessage;
import software.amazon.awssdk.services.cloudwatch.model.Datapoint;
import software.amazon.awssdk.services.cloudwatch.model.DeleteAlarmsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DeleteAnomalyDetectorRequest;
import software.amazon.awssdk.services.cloudwatch.model.DeleteDashboardsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmHistoryRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmHistoryResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsForMetricRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsForMetricResponse;
import software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsRequest;
import software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAnomalyDetectorsRequest;
```

```
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAnomalyDetectorsResponse;
import software.amazon.awssdk.services.cloudwatch.model.Dimension;
import software.amazon.awssdk.services.cloudwatch.model.GetMetricDataRequest;
import software.amazon.awssdk.services.cloudwatch.model.GetMetricDataResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricStatisticsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricStatisticsResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricWidgetImageRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricWidgetImageResponse;
import software.amazon.awssdk.services.cloudwatch.model.HistoryItemType;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsRequest;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsResponse;
import software.amazon.awssdk.services.cloudwatch.model.Metric;
import software.amazon.awssdk.services.cloudwatch.model.MetricAlarm;
import software.amazon.awssdk.services.cloudwatch.model.MetricDataQuery;
import software.amazon.awssdk.services.cloudwatch.model.MetricDataResult;
import software.amazon.awssdk.services.cloudwatch.model.MetricDatum;
import software.amazon.awssdk.services.cloudwatch.model.MetricStat;
import
 software.amazon.awssdk.services.cloudwatch.model.PutAnomalyDetectorRequest;
import software.amazon.awssdk.services.cloudwatch.model.PutDashboardRequest;
import software.amazon.awssdk.services.cloudwatch.model.PutDashboardResponse;
import software.amazon.awssdk.services.cloudwatch.model.PutMetricAlarmRequest;
import software.amazon.awssdk.services.cloudwatch.model.PutMetricDataRequest;
import software.amazon.awssdk.services.cloudwatch.model.ScanBy;
import
 software.amazon.awssdk.services.cloudwatch.model.SingleMetricAnomalyDetector;
import software.amazon.awssdk.services.cloudwatch.model.StandardUnit;
import software.amazon.awssdk.services.cloudwatch.model.Statistic;
import
 software.amazon.awssdk.services.cloudwatch.paginators.ListDashboardsIterable;
import software.amazon.awssdk.services.cloudwatch.paginators.ListMetricsIterable;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.time.Instant;
```

```
import java.time.ZoneOffset;
import java.time.ZonedDateTime;
import java.time.format.DateTimeFormatter;
import java.time.temporal.ChronoUnit;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 * To enable billing metrics and statistics for this example, make sure billing
 * alerts are enabled for your account:
 * https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
monitor_estimated_charges_with_cloudwatch.html#turning_on_billing_metrics
 *
 * This Java code example performs the following tasks:
 *
 * 1. List available namespaces from Amazon CloudWatch.
 * 2. List available metrics within the selected Namespace.
 * 3. Get statistics for the selected metric over the last day.
 * 4. Get CloudWatch estimated billing for the last week.
 * 5. Create a new CloudWatch dashboard with metrics.
 * 6. List dashboards using a paginator.
 * 7. Create a new custom metric by adding data for it.
 * 8. Add the custom metric to the dashboard.
 * 9. Create an alarm for the custom metric.
 * 10. Describe current alarms.
 * 11. Get current data for the new custom metric.
 * 12. Push data into the custom metric to trigger the alarm.
 * 13. Check the alarm state using the action DescribeAlarmsForMetric.
 * 14. Get alarm history for the new alarm.
 * 15. Add an anomaly detector for the custom metric.
 * 16. Describe current anomaly detectors.
 * 17. Get a metric image for the custom metric.
 * 18. Clean up the Amazon CloudWatch resources.
 */
public class CloudWatchScenario {
```

```
   public static final String DASHES = new String(new char[80]).replace("\0",
"-");

   public static void main(String[] args) throws IOException {
       final String usage = """

               Usage:
                 <myDate> <costDateWeek> <dashboardName> <dashboardJson>
<dashboardAdd> <settings> <metricImage> \s

               Where:
                 myDate - The start date to use to get metric statistics. (For
example, 2023-01-11T18:35:24.00Z.)\s
                 costDateWeek - The start date to use to get AWS/Billinget
statistics. (For example, 2023-01-11T18:35:24.00Z.)\s
                 dashboardName - The name of the dashboard to create.\s
                 dashboardJson - The location of a JSON file to use to create a
dashboard. (See Readme file.)\s
                 dashboardAdd - The location of a JSON file to use to update a
dashboard. (See Readme file.)\s
                 settings - The location of a JSON file from which various
values are read. (See Readme file.)\s
                 metricImage - The location of a BMP file that is used to create
a graph.\s
               """;

       if (args.length != 7) {
           System.out.println(usage);
           System.exit(1);
       }

       Region region = Region.US_EAST_1;
       String myDate = args[0];
       String costDateWeek = args[1];
       String dashboardName = args[2];
       String dashboardJson = args[3];
       String dashboardAdd = args[4];
       String settings = args[5];
       String metricImage = args[6];

       Double dataPoint = Double.parseDouble("10.0");
       Scanner sc = new Scanner(System.in);
       CloudWatchClient cw = CloudWatchClient.builder()
               .region(region)
```

```
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon CloudWatch example scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(
                "1. List at least five available unique namespaces from Amazon
CloudWatch. Select one from the list.");
        ArrayList<String> list = listNameSpaces(cw);
        for (int z = 0; z < 5; z++) {
            int index = z + 1;
            System.out.println("    " + index + ". " + list.get(z));
        }

        String selectedNamespace = "";
        String selectedMetrics = "";
        int num = Integer.parseInt(sc.nextLine());
        if (1 <= num && num <= 5) {
            selectedNamespace = list.get(num - 1);
        } else {
            System.out.println("You did not select a valid option.");
            System.exit(1);
        }
        System.out.println("You selected " + selectedNamespace);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("2. List available metrics within the selected
namespace and select one from the list.");
        ArrayList<String> metList = listMets(cw, selectedNamespace);
        for (int z = 0; z < 5; z++) {
            int index = z + 1;
            System.out.println("    " + index + ". " + metList.get(z));
        }
        num = Integer.parseInt(sc.nextLine());
        if (1 <= num && num <= 5) {
            selectedMetrics = metList.get(num - 1);
        } else {
            System.out.println("You did not select a valid option.");
            System.exit(1);
        }
```

```
        System.out.println("You selected " + selectedMetrics);
        Dimension myDimension = getSpecificMet(cw, selectedNamespace);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("3. Get statistics for the selected metric over the
last day.");
        String metricOption = "";
        ArrayList<String> statTypes = new ArrayList<>();
        statTypes.add("SampleCount");
        statTypes.add("Average");
        statTypes.add("Sum");
        statTypes.add("Minimum");
        statTypes.add("Maximum");

        for (int t = 0; t < 5; t++) {
            System.out.println("    " + (t + 1) + ". " + statTypes.get(t));
        }
        System.out.println("Select a metric statistic by entering a number from
the preceding list:");
        num = Integer.parseInt(sc.nextLine());
        if (1 <= num && num <= 5) {
            metricOption = statTypes.get(num - 1);
        } else {
            System.out.println("You did not select a valid option.");
            System.exit(1);
        }
        System.out.println("You selected " + metricOption);
        getAndDisplayMetricStatistics(cw, selectedNamespace, selectedMetrics,
metricOption, myDate, myDimension);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. Get CloudWatch estimated billing for the last
week.");
        getMetricStatistics(cw, costDateWeek);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("5. Create a new CloudWatch dashboard with metrics.");
        createDashboardWithMetrics(cw, dashboardName, dashboardJson);
        System.out.println(DASHES);

        System.out.println(DASHES);
```

```
        System.out.println("6. List dashboards using a paginator.");
        listDashboards(cw);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("7. Create a new custom metric by adding data to
it.");
        createNewCustomMetric(cw, dataPoint);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("8. Add an additional metric to the dashboard.");
        addMetricToDashboard(cw, dashboardAdd, dashboardName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("9. Create an alarm for the custom metric.");
        String alarmName = createAlarm(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("10. Describe ten current alarms.");
        describeAlarms(cw);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("11. Get current data for new custom metric.");
        getCustomMetricData(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("12. Push data into the custom metric to trigger the
alarm.");
        addMetricDataForAlarm(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("13. Check the alarm state using the action
DescribeAlarmsForMetric.");
        checkForMetricAlarm(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("14. Get alarm history for the new alarm.");
```

```
        getAlarmHistory(cw, settings, myDate);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("15. Add an anomaly detector for the custom metric.");
        addAnomalyDetector(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("16. Describe current anomaly detectors.");
        describeAnomalyDetectors(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("17. Get a metric image for the custom metric.");
        getAndOpenMetricImage(cw, metricImage);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("18. Clean up the Amazon CloudWatch resources.");
        deleteDashboard(cw, dashboardName);
        deleteCWAlarm(cw, alarmName);
        deleteAnomalyDetector(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("The Amazon CloudWatch example scenario is
complete.");
        System.out.println(DASHES);
        cw.close();
    }

    public static void deleteAnomalyDetector(CloudWatchClient cw, String
fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();
```

```
            SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .stat("Maximum")
                    .build();

            DeleteAnomalyDetectorRequest request =
DeleteAnomalyDetectorRequest.builder()
                    .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                    .build();

            cw.deleteAnomalyDetector(request);
            System.out.println("Successfully deleted the Anomaly Detector.");

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public static void deleteCWAlarm(CloudWatchClient cw, String alarmName) {
        try {
            DeleteAlarmsRequest request = DeleteAlarmsRequest.builder()
                    .alarmNames(alarmName)
                    .build();

            cw.deleteAlarms(request);
            System.out.println("Successfully deleted alarm " + alarmName);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void deleteDashboard(CloudWatchClient cw, String dashboardName)
{
        try {
            DeleteDashboardsRequest dashboardsRequest =
DeleteDashboardsRequest.builder()
```

```
                    .dashboardNames(dashboardName)
                    .build();
            cw.deleteDashboards(dashboardsRequest);
            System.out.println(dashboardName + " was successfully deleted.");

        } catch (CloudWatchException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void getAndOpenMetricImage(CloudWatchClient cw, String
fileName) {
        System.out.println("Getting Image data for custom metric.");
        try {
            String myJSON = "{\n" +
                    "  \"title\": \"Example Metric Graph\",\n" +
                    "  \"view\": \"timeSeries\",\n" +
                    "  \"stacked \": false,\n" +
                    "  \"period\": 10,\n" +
                    "  \"width\": 1400,\n" +
                    "  \"height\": 600,\n" +
                    "  \"metrics\": [\n" +
                    "    [\n" +
                    "      \"AWS/Billing\",\n" +
                    "      \"EstimatedCharges\",\n" +
                    "      \"Currency\",\n" +
                    "      \"USD\"\n" +
                    "    ]\n" +
                    "  ]\n" +
                    "}";

            GetMetricWidgetImageRequest imageRequest =
GetMetricWidgetImageRequest.builder()
                    .metricWidget(myJSON)
                    .build();

            GetMetricWidgetImageResponse response =
cw.getMetricWidgetImage(imageRequest);
            SdkBytes sdkBytes = response.metricWidgetImage();
            byte[] bytes = sdkBytes.asByteArray();
            File outputFile = new File(fileName);
            try (FileOutputStream outputStream = new
FileOutputStream(outputFile)) {
```

```
                outputStream.write(bytes);
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void describeAnomalyDetectors(CloudWatchClient cw, String
fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();
            DescribeAnomalyDetectorsRequest detectorsRequest =
DescribeAnomalyDetectorsRequest.builder()
                    .maxResults(10)
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .build();

            DescribeAnomalyDetectorsResponse response =
cw.describeAnomalyDetectors(detectorsRequest);
            List<AnomalyDetector> anomalyDetectorList =
response.anomalyDetectors();
            for (AnomalyDetector detector : anomalyDetectorList) {
                System.out.println("Metric name: " +
detector.singleMetricAnomalyDetector().metricName());
                System.out.println("State: " + detector.stateValue());
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

```
   public static void addAnomalyDetector(CloudWatchClient cw, String fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();

           SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .stat("Maximum")
                   .build();

           PutAnomalyDetectorRequest anomalyDetectorRequest =
PutAnomalyDetectorRequest.builder()
                   .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                   .build();

           cw.putAnomalyDetector(anomalyDetectorRequest);
           System.out.println("Added anomaly detector for metric " +
customMetricName + ".");

       } catch (CloudWatchException | IOException e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
   }

   public static void getAlarmHistory(CloudWatchClient cw, String fileName,
String date) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String alarmName = rootNode.findValue("exampleAlarmName").asText();
```

```
            Instant start = Instant.parse(date);
            Instant endDate = Instant.now();
            DescribeAlarmHistoryRequest historyRequest =
DescribeAlarmHistoryRequest.builder()
                    .startDate(start)
                    .endDate(endDate)
                    .alarmName(alarmName)
                    .historyItemType(HistoryItemType.ACTION)
                    .build();

            DescribeAlarmHistoryResponse response =
cw.describeAlarmHistory(historyRequest);
            List<AlarmHistoryItem> historyItems = response.alarmHistoryItems();
            if (historyItems.isEmpty()) {
                System.out.println("No alarm history data found for " + alarmName
+ ".");
            } else {
                for (AlarmHistoryItem item : historyItems) {
                    System.out.println("History summary: " +
item.historySummary());
                    System.out.println("Time stamp: " + item.timestamp());
                }
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void checkForMetricAlarm(CloudWatchClient cw, String fileName)
{
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();
            boolean hasAlarm = false;
            int retries = 10;
```

```
            DescribeAlarmsForMetricRequest metricRequest =
DescribeAlarmsForMetricRequest.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .build();

            while (!hasAlarm && retries > 0) {
                DescribeAlarmsForMetricResponse response =
cw.describeAlarmsForMetric(metricRequest);
                hasAlarm = response.hasMetricAlarms();
                retries--;
                Thread.sleep(20000);
                System.out.println(".");
            }
            if (!hasAlarm)
                System.out.println("No Alarm state found for " + customMetricName
+ " after 10 retries.");
            else
                System.out.println("Alarm state found for " + customMetricName +
".");

        } catch (CloudWatchException | IOException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void addMetricDataForAlarm(CloudWatchClient cw, String
fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            // Set an Instant object.
            String time =
ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT);
```

```
            Instant instant = Instant.parse(time);

            MetricDatum datum = MetricDatum.builder()
                    .metricName(customMetricName)
                    .unit(StandardUnit.NONE)
                    .value(1001.00)
                    .timestamp(instant)
                    .build();

            MetricDatum datum2 = MetricDatum.builder()
                    .metricName(customMetricName)
                    .unit(StandardUnit.NONE)
                    .value(1002.00)
                    .timestamp(instant)
                    .build();

            List<MetricDatum> metricDataList = new ArrayList<>();
            metricDataList.add(datum);
            metricDataList.add(datum2);

            PutMetricDataRequest request = PutMetricDataRequest.builder()
                    .namespace(customMetricNamespace)
                    .metricData(metricDataList)
                    .build();

            cw.putMetricData(request);
            System.out.println("Added metric values for for metric " +
customMetricName);

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void getCustomMetricData(CloudWatchClient cw, String fileName)
{
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
```

```
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            // Set the date.
            Instant nowDate = Instant.now();

            long hours = 1;
            long minutes = 30;
            Instant date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(minutes,
                    ChronoUnit.MINUTES);

            Metric met = Metric.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .build();

            MetricStat metStat = MetricStat.builder()
                    .stat("Maximum")
                    .period(1)
                    .metric(met)
                    .build();

            MetricDataQuery dataQUery = MetricDataQuery.builder()
                    .metricStat(metStat)
                    .id("foo2")
                    .returnData(true)
                    .build();

            List<MetricDataQuery> dq = new ArrayList<>();
            dq.add(dataQUery);

            GetMetricDataRequest getMetReq = GetMetricDataRequest.builder()
                    .maxDatapoints(10)
                    .scanBy(ScanBy.TIMESTAMP_DESCENDING)
                    .startTime(nowDate)
                    .endTime(date2)
                    .metricDataQueries(dq)
                    .build();

            GetMetricDataResponse response = cw.getMetricData(getMetReq);
            List<MetricDataResult> data = response.metricDataResults();
            for (MetricDataResult item : data) {
```

```
                System.out.println("The label is " + item.label());
                System.out.println("The status code is " +
item.statusCode().toString());
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void describeAlarms(CloudWatchClient cw) {
        try {
            List<AlarmType> typeList = new ArrayList<>();
            typeList.add(AlarmType.METRIC_ALARM);

            DescribeAlarmsRequest alarmsRequest = DescribeAlarmsRequest.builder()
                    .alarmTypes(typeList)
                    .maxRecords(10)
                    .build();

            DescribeAlarmsResponse response = cw.describeAlarms(alarmsRequest);
            List<MetricAlarm> alarmList = response.metricAlarms();
            for (MetricAlarm alarm : alarmList) {
                System.out.println("Alarm name: " + alarm.alarmName());
                System.out.println("Alarm description: " +
alarm.alarmDescription());
            }
        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static String createAlarm(CloudWatchClient cw, String fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
```

```
            String customMetricName =
rootNode.findValue("customMetricName").asText();
            String alarmName = rootNode.findValue("exampleAlarmName").asText();
            String emailTopic = rootNode.findValue("emailTopic").asText();
            String accountId = rootNode.findValue("accountId").asText();
            String region = rootNode.findValue("region").asText();

            // Create a List for alarm actions.
            List<String> alarmActions = new ArrayList<>();
            alarmActions.add("arn:aws:sns:" + region + ":" + accountId + ":" +
emailTopic);
            PutMetricAlarmRequest alarmRequest = PutMetricAlarmRequest.builder()
                    .alarmActions(alarmActions)
                    .alarmDescription("Example metric alarm")
                    .alarmName(alarmName)

.comparisonOperator(ComparisonOperator.GREATER_THAN_OR_EQUAL_TO_THRESHOLD)
                    .threshold(100.00)
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .evaluationPeriods(1)
                    .period(10)
                    .statistic("Maximum")
                    .datapointsToAlarm(1)
                    .treatMissingData("ignore")
                    .build();

            cw.putMetricAlarm(alarmRequest);
            System.out.println(alarmName + " was successfully created!");
            return alarmName;

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
        return "";
    }

    public static void addMetricToDashboard(CloudWatchClient cw, String fileName,
String dashboardName) {
        try {
            PutDashboardRequest dashboardRequest = PutDashboardRequest.builder()
                    .dashboardName(dashboardName)
                    .dashboardBody(readFileAsString(fileName))
```

```
                    .build();

            cw.putDashboard(dashboardRequest);
            System.out.println(dashboardName + " was successfully updated.");

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void createNewCustomMetric(CloudWatchClient cw, Double
dataPoint) {
        try {
            Dimension dimension = Dimension.builder()
                    .name("UNIQUE_PAGES")
                    .value("URLS")
                    .build();

            // Set an Instant object.
            String time =
ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT);
            Instant instant = Instant.parse(time);

            MetricDatum datum = MetricDatum.builder()
                    .metricName("PAGES_VISITED")
                    .unit(StandardUnit.NONE)
                    .value(dataPoint)
                    .timestamp(instant)
                    .dimensions(dimension)
                    .build();

            PutMetricDataRequest request = PutMetricDataRequest.builder()
                    .namespace("SITE/TRAFFIC")
                    .metricData(datum)
                    .build();

            cw.putMetricData(request);
            System.out.println("Added metric values for for metric
PAGES_VISITED");

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
```

```
        }
    }

    public static void listDashboards(CloudWatchClient cw) {
        try {
            ListDashboardsIterable listRes = cw.listDashboardsPaginator();
            listRes.stream()
                    .flatMap(r -> r.dashboardEntries().stream())
                    .forEach(entry -> {
                        System.out.println("Dashboard name is: " +
entry.dashboardName());
                        System.out.println("Dashboard ARN is: " +
entry.dashboardArn());
                    });

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void createDashboardWithMetrics(CloudWatchClient cw, String
dashboardName, String fileName) {
        try {
            PutDashboardRequest dashboardRequest = PutDashboardRequest.builder()
                    .dashboardName(dashboardName)
                    .dashboardBody(readFileAsString(fileName))
                    .build();

            PutDashboardResponse response = cw.putDashboard(dashboardRequest);
            System.out.println(dashboardName + " was successfully created.");
            List<DashboardValidationMessage> messages =
response.dashboardValidationMessages();
            if (messages.isEmpty()) {
                System.out.println("There are no messages in the new Dashboard");
            } else {
                for (DashboardValidationMessage message : messages) {
                    System.out.println("Message is: " + message.message());
                }
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
```

```
        }
    }

    public static String readFileAsString(String file) throws IOException {
        return new String(Files.readAllBytes(Paths.get(file)));
    }

    public static void getMetricStatistics(CloudWatchClient cw, String
costDateWeek) {
        try {
            Instant start = Instant.parse(costDateWeek);
            Instant endDate = Instant.now();
            Dimension dimension = Dimension.builder()
                    .name("Currency")
                    .value("USD")
                    .build();

            List<Dimension> dimensionList = new ArrayList<>();
            dimensionList.add(dimension);
            GetMetricStatisticsRequest statisticsRequest =
GetMetricStatisticsRequest.builder()
                    .metricName("EstimatedCharges")
                    .namespace("AWS/Billing")
                    .dimensions(dimensionList)
                    .statistics(Statistic.MAXIMUM)
                    .startTime(start)
                    .endTime(endDate)
                    .period(86400)
                    .build();

            GetMetricStatisticsResponse response =
cw.getMetricStatistics(statisticsRequest);
            List<Datapoint> data = response.datapoints();
            if (!data.isEmpty()) {
                for (Datapoint datapoint : data) {
                    System.out
                            .println("Timestamp: " + datapoint.timestamp() + "
Maximum value: " + datapoint.maximum());
                }
            } else {
                System.out.println("The returned data list is empty");
            }

        } catch (CloudWatchException e) {
```

```
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void getAndDisplayMetricStatistics(CloudWatchClient cw, String
nameSpace, String metVal,
            String metricOption, String date, Dimension myDimension) {
        try {
            Instant start = Instant.parse(date);
            Instant endDate = Instant.now();

            GetMetricStatisticsRequest statisticsRequest =
GetMetricStatisticsRequest.builder()
                    .endTime(endDate)
                    .startTime(start)
                    .dimensions(myDimension)
                    .metricName(metVal)
                    .namespace(nameSpace)
                    .period(86400)
                    .statistics(Statistic.fromValue(metricOption))
                    .build();

            GetMetricStatisticsResponse response =
cw.getMetricStatistics(statisticsRequest);
            List<Datapoint> data = response.datapoints();
            if (!data.isEmpty()) {
                for (Datapoint datapoint : data) {
                    System.out
                            .println("Timestamp: " + datapoint.timestamp() + "
Maximum value: " + datapoint.maximum());
                }
            } else {
                System.out.println("The returned data list is empty");
            }

        } catch (CloudWatchException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static Dimension getSpecificMet(CloudWatchClient cw, String namespace)
{
```

```
        try {
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .namespace(namespace)
                    .build();

            ListMetricsResponse response = cw.listMetrics(request);
            List<Metric> myList = response.metrics();
            Metric metric = myList.get(0);
            return metric.dimensions().get(0);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    public static ArrayList<String> listMets(CloudWatchClient cw, String
namespace) {
        try {
            ArrayList<String> metList = new ArrayList<>();
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .namespace(namespace)
                    .build();

            ListMetricsIterable listRes = cw.listMetricsPaginator(request);
            listRes.stream()
                    .flatMap(r -> r.metrics().stream())
                    .forEach(metrics -> metList.add(metrics.metricName()));

            return metList;

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    public static ArrayList<String> listNameSpaces(CloudWatchClient cw) {
        try {
            ArrayList<String> nameSpaceList = new ArrayList<>();
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .build();
```

```
            ListMetricsIterable listRes = cw.listMetricsPaginator(request);
            listRes.stream()
                    .flatMap(r -> r.metrics().stream())
                    .forEach(metrics -> {
                        String data = metrics.namespace();
                        if (!nameSpaceList.contains(data)) {
                            nameSpaceList.add(data);
                        }
                    });

            return nameSpaceList;
        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
}
```

- API-Details finden Sie in den folgenden Themen der AWS SDK for Java 2.x -API-Referenz.
  - DeleteAlarms
  - DeleteAnomalyDetector
  - DeleteDashboards
  - DescribeAlarmHistory
  - DescribeAlarms
  - DescribeAlarmsForMetric
  - DescribeAnomalyDetectors
  - GetMetricData
  - GetMetricStatistics
  - GetMetricWidgetImage
  - ListMetrics
  - PutAnomalyDetector
  - PutDashboard
  - PutMetricAlarm
  - PutMetricData

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
/**
 Before running this Kotlin code example, set up your development environment,
 including your credentials.

 For more information, see the following documentation topic:
 https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html

 To enable billing metrics and statistics for this example, make sure billing
 alerts are enabled for your account:
 https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
monitor_estimated_charges_with_cloudwatch.html#turning_on_billing_metrics

 This Kotlin code example performs the following tasks:

 1. List available namespaces from Amazon CloudWatch. Select a namespace from the
 list.
 2. List available metrics within the selected namespace.
 3. Get statistics for the selected metric over the last day.
 4. Get CloudWatch estimated billing for the last week.
 5. Create a new CloudWatch dashboard with metrics.
 6. List dashboards using a paginator.
 7. Create a new custom metric by adding data for it.
 8. Add the custom metric to the dashboard.
 9. Create an alarm for the custom metric.
 10. Describe current alarms.
 11. Get current data for the new custom metric.
 12. Push data into the custom metric to trigger the alarm.
 13. Check the alarm state using the action DescribeAlarmsForMetric.
 14. Get alarm history for the new alarm.
 15. Add an anomaly detector for the custom metric.
 16. Describe current anomaly detectors.
 17. Get a metric image for the custom metric.
```

```
 18. Clean up the Amazon CloudWatch resources.
 */

val DASHES: String? = String(CharArray(80)).replace("\u0000", "-")
suspend fun main(args: Array<String>) {
    val usage = """
        Usage:
            <myDate> <costDateWeek> <dashboardName> <dashboardJson>
 <dashboardAdd> <settings> <metricImage>

        Where:
            myDate - The start date to use to get metric statistics. (For
 example, 2023-01-11T18:35:24.00Z.)
            costDateWeek - The start date to use to get AWS Billing and Cost
 Management statistics. (For example, 2023-01-11T18:35:24.00Z.)
            dashboardName - The name of the dashboard to create.
            dashboardJson - The location of a JSON file to use to create a
 dashboard. (See Readme file.)
            dashboardAdd - The location of a JSON file to use to update a
 dashboard. (See Readme file.)
            settings - The location of a JSON file from which various values are
 read. (See Readme file.)
            metricImage - The location of a BMP file that is used to create a
 graph.
    """

    if (args.size != 7) {
        println(usage)
        System.exit(1)
    }

    val myDate = args[0]
    val costDateWeek = args[1]
    val dashboardName = args[2]
    val dashboardJson = args[3]
    val dashboardAdd = args[4]
    val settings = args[5]
    var metricImage = args[6]
    val dataPoint = "10.0".toDouble()
    val inOb = Scanner(System.`in`)

    println(DASHES)
    println("Welcome to the Amazon CloudWatch example scenario.")
    println(DASHES)
```

```
    println(DASHES)
    println("1. List at least five available unique namespaces from Amazon
CloudWatch. Select a CloudWatch namespace from the list.")
    val list: ArrayList<String> = listNameSpaces()
    for (z in 0..4) {
        println("    ${z + 1}. ${list[z]}")
    }

    var selectedNamespace: String
    var selectedMetrics = ""
    var num = inOb.nextLine().toInt()
    println("You selected $num")

    if (1 <= num && num <= 5) {
        selectedNamespace = list[num - 1]
    } else {
        println("You did not select a valid option.")
        exitProcess(1)
    }
    println("You selected $selectedNamespace")
    println(DASHES)

    println(DASHES)
    println("2. List available metrics within the selected namespace and select
one from the list.")
    val metList = listMets(selectedNamespace)
    for (z in 0..4) {
        println("    ${ z + 1}. ${metList?.get(z)}")
    }
    num = inOb.nextLine().toInt()
    if (1 <= num && num <= 5) {
        selectedMetrics = metList!![num - 1]
    } else {
        println("You did not select a valid option.")
        System.exit(1)
    }
    println("You selected $selectedMetrics")
    val myDimension = getSpecificMet(selectedNamespace)
    if (myDimension == null) {
        println("Error - Dimension is null")
        exitProcess(1)
    }
    println(DASHES)
```

```
    println(DASHES)
    println("3. Get statistics for the selected metric over the last day.")
    val metricOption: String
    val statTypes = ArrayList<String>()
    statTypes.add("SampleCount")
    statTypes.add("Average")
    statTypes.add("Sum")
    statTypes.add("Minimum")
    statTypes.add("Maximum")

    for (t in 0..4) {
        println("    ${t + 1}. ${statTypes[t]}")
    }
    println("Select a metric statistic by entering a number from the preceding
list:")
    num = inOb.nextLine().toInt()
    if (1 <= num && num <= 5) {
        metricOption = statTypes[num - 1]
    } else {
        println("You did not select a valid option.")
        exitProcess(1)
    }
    println("You selected $metricOption")
    getAndDisplayMetricStatistics(selectedNamespace, selectedMetrics,
metricOption, myDate, myDimension)
    println(DASHES)

    println(DASHES)
    println("4. Get CloudWatch estimated billing for the last week.")
    getMetricStatistics(costDateWeek)
    println(DASHES)

    println(DASHES)
    println("5. Create a new CloudWatch dashboard with metrics.")
    createDashboardWithMetrics(dashboardName, dashboardJson)
    println(DASHES)

    println(DASHES)
    println("6. List dashboards using a paginator.")
    listDashboards()
    println(DASHES)

    println(DASHES)
```

```
    println("7. Create a new custom metric by adding data to it.")
    createNewCustomMetric(dataPoint)
    println(DASHES)

    println(DASHES)
    println("8. Add an additional metric to the dashboard.")
    addMetricToDashboard(dashboardAdd, dashboardName)
    println(DASHES)

    println(DASHES)
    println("9. Create an alarm for the custom metric.")
    val alarmName: String = createAlarm(settings)
    println(DASHES)

    println(DASHES)
    println("10. Describe 10 current alarms.")
    describeAlarms()
    println(DASHES)

    println(DASHES)
    println("11. Get current data for the new custom metric.")
    getCustomMetricData(settings)
    println(DASHES)

    println(DASHES)
    println("12. Push data into the custom metric to trigger the alarm.")
    addMetricDataForAlarm(settings)
    println(DASHES)

    println(DASHES)
    println("13. Check the alarm state using the action
DescribeAlarmsForMetric.")
    checkForMetricAlarm(settings)
    println(DASHES)

    println(DASHES)
    println("14. Get alarm history for the new alarm.")
    getAlarmHistory(settings, myDate)
    println(DASHES)

    println(DASHES)
    println("15. Add an anomaly detector for the custom metric.")
    addAnomalyDetector(settings)
    println(DASHES)
```

```
    println(DASHES)
    println("16. Describe current anomaly detectors.")
    describeAnomalyDetectors(settings)
    println(DASHES)

    println(DASHES)
    println("17. Get a metric image for the custom metric.")
    getAndOpenMetricImage(metricImage)
    println(DASHES)

    println(DASHES)
    println("18. Clean up the Amazon CloudWatch resources.")
    deleteDashboard(dashboardName)
    deleteAlarm(alarmName)
    deleteAnomalyDetector(settings)
    println(DASHES)

    println(DASHES)
    println("The Amazon CloudWatch example scenario is complete.")
    println(DASHES)
}

suspend fun deleteAnomalyDetector(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val request = DeleteAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAnomalyDetector(request)
        println("Successfully deleted the Anomaly Detector.")
```

```
    }
}

suspend fun deleteAlarm(alarmNameVal: String) {
    val request = DeleteAlarmsRequest {
        alarmNames = listOf(alarmNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAlarms(request)
        println("Successfully deleted alarm $alarmNameVal")
    }
}

suspend fun deleteDashboard(dashboardName: String) {
    val dashboardsRequest = DeleteDashboardsRequest {
        dashboardNames = listOf(dashboardName)
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteDashboards(dashboardsRequest)
        println("$dashboardName was successfully deleted.")
    }
}

suspend fun getAndOpenMetricImage(fileName: String) {
    println("Getting Image data for custom metric.")
    val myJSON = """{
        "title": "Example Metric Graph",
        "view": "timeSeries",
        "stacked ": false,
        "period": 10,
        "width": 1400,
        "height": 600,
        "metrics": [
            [
            "AWS/Billing",
            "EstimatedCharges",
            "Currency",
            "USD"
            ]
        ]
        }"""

    val imageRequest = GetMetricWidgetImageRequest {
```

```
        metricWidget = myJSON
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricWidgetImage(imageRequest)
        val bytes = response.metricWidgetImage
        if (bytes != null) {
            File(fileName).writeBytes(bytes)
        }
    }
    println("You have successfully written data to $fileName")
}

suspend fun describeAnomalyDetectors(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val detectorsRequest = DescribeAnomalyDetectorsRequest {
        maxResults = 10
        metricName = customMetricName
        namespace = customMetricNamespace
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAnomalyDetectors(detectorsRequest)
        response.anomalyDetectors?.forEach { detector ->
            println("Metric name:
 ${detector.singleMetricAnomalyDetector?.metricName}")
            println("State: ${detector.stateValue}")
        }
    }
}

suspend fun addAnomalyDetector(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()
```

```
    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val anomalyDetectorRequest = PutAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putAnomalyDetector(anomalyDetectorRequest)
        println("Added anomaly detector for metric $customMetricName.")
    }
}

suspend fun getAlarmHistory(fileName: String, date: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val alarmNameVal = rootNode.findValue("exampleAlarmName").asText()
    val start = Instant.parse(date)
    val endDateVal = Instant.now()

    val historyRequest = DescribeAlarmHistoryRequest {
        startDate = aws.smithy.kotlin.runtime.time.Instant(start)
        endDate = aws.smithy.kotlin.runtime.time.Instant(endDateVal)
        alarmName = alarmNameVal
        historyItemType = HistoryItemType.Action
    }

    CloudWatchClient { credentialsProvider = EnvironmentCredentialsProvider();
 region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarmHistory(historyRequest)
        val historyItems = response.alarmHistoryItems
        if (historyItems != null) {
            if (historyItems.isEmpty()) {
                println("No alarm history data found for $alarmNameVal.")
            } else {
                for (item in historyItems) {
                    println("History summary ${item.historySummary}")
                    println("Time stamp: ${item.timestamp}")
                }
            }
```

```
        }
    }
}

suspend fun checkForMetricAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()
    var hasAlarm = false
    var retries = 10

    val metricRequest = DescribeAlarmsForMetricRequest {
        metricName = customMetricName
        namespace = customMetricNamespace
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        while (!hasAlarm && retries > 0) {
            val response = cwClient.describeAlarmsForMetric(metricRequest)
            if (response.metricAlarms?.count()!! > 0) {
                hasAlarm = true
            }
            retries--
            delay(20000)
            println(".")
        }
        if (!hasAlarm) println("No Alarm state found for $customMetricName after
 10 retries.") else println("Alarm state found for $customMetricName.")
    }
}

suspend fun addMetricDataForAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set an Instant object.
    val time =
 ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT)
```

```
    val instant = Instant.parse(time)
    val datum = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1001.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }

    val datum2 = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1002.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }

    val metricDataList = ArrayList<MetricDatum>()
    metricDataList.add(datum)
    metricDataList.add(datum2)

    val request = PutMetricDataRequest {
        namespace = customMetricNamespace
        metricData = metricDataList
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricData(request)
        println("Added metric values for for metric $customMetricName")
    }
}

suspend fun getCustomMetricData(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set the date.
    val nowDate = Instant.now()
    val hours: Long = 1
    val minutes: Long = 30
    val date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(
        minutes,
```

```
        ChronoUnit.MINUTES
    )

    val met = Metric {
        metricName = customMetricName
        namespace = customMetricNamespace
    }

    val metStat = MetricStat {
        stat = "Maximum"
        period = 1
        metric = met
    }

    val dataQUery = MetricDataQuery {
        metricStat = metStat
        id = "foo2"
        returnData = true
    }

    val dq = ArrayList<MetricDataQuery>()
    dq.add(dataQUery)
    val getMetReq = GetMetricDataRequest {
        maxDatapoints = 10
        scanBy = ScanBy.TimestampDescending
        startTime = aws.smithy.kotlin.runtime.time.Instant(nowDate)
        endTime = aws.smithy.kotlin.runtime.time.Instant(date2)
        metricDataQueries = dq
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricData(getMetReq)
        response.metricDataResults?.forEach { item ->
            println("The label is ${item.label}")
            println("The status code is ${item.statusCode}")
        }
    }
}

suspend fun describeAlarms() {
    val typeList = ArrayList<AlarmType>()
    typeList.add(AlarmType.MetricAlarm)
    val alarmsRequest = DescribeAlarmsRequest {
        alarmTypes = typeList
```

```
        maxRecords = 10
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarms(alarmsRequest)
        response.metricAlarms?.forEach { alarm ->
            println("Alarm name: ${alarm.alarmName}")
            println("Alarm description: ${alarm.alarmDescription}")
        }
    }
}

suspend fun createAlarm(fileName: String): String {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode: JsonNode = ObjectMapper().readTree(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()
    val alarmNameVal = rootNode.findValue("exampleAlarmName").asText()
    val emailTopic = rootNode.findValue("emailTopic").asText()
    val accountId = rootNode.findValue("accountId").asText()
    val region2 = rootNode.findValue("region").asText()

    // Create a List for alarm actions.
    val alarmActionObs: MutableList<String> = ArrayList()
    alarmActionObs.add("arn:aws:sns:$region2:$accountId:$emailTopic")
    val alarmRequest = PutMetricAlarmRequest {
        alarmActions = alarmActionObs
        alarmDescription = "Example metric alarm"
        alarmName = alarmNameVal
        comparisonOperator = ComparisonOperator.GreaterThanOrEqualToThreshold
        threshold = 100.00
        metricName = customMetricName
        namespace = customMetricNamespace
        evaluationPeriods = 1
        period = 10
        statistic = Statistic.Maximum
        datapointsToAlarm = 1
        treatMissingData = "ignore"
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricAlarm(alarmRequest)
```

```
        println("$alarmNameVal was successfully created!")
        return alarmNameVal
    }
}

suspend fun addMetricToDashboard(fileNameVal: String, dashboardNameVal: String) {
    val dashboardRequest = PutDashboardRequest {
        dashboardName = dashboardNameVal
        dashboardBody = readFileAsString(fileNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putDashboard(dashboardRequest)
        println("$dashboardNameVal was successfully updated.")
    }
}

suspend fun createNewCustomMetric(dataPoint: Double) {
    val dimension = Dimension {
        name = "UNIQUE_PAGES"
        value = "URLS"
    }

    // Set an Instant object.
    val time =
 ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT)
    val instant = Instant.parse(time)
    val datum = MetricDatum {
        metricName = "PAGES_VISITED"
        unit = StandardUnit.None
        value = dataPoint
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
        dimensions = listOf(dimension)
    }

    val request = PutMetricDataRequest {
        namespace = "SITE/TRAFFIC"
        metricData = listOf(datum)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricData(request)
        println("Added metric values for for metric PAGES_VISITED")
    }
```

```
}

suspend fun listDashboards() {
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.listDashboardsPaginated({})
            .transform { it.dashboardEntries?.forEach { obj -> emit(obj) } }
            .collect { obj ->
                println("Name is ${obj.dashboardName}")
                println("Dashboard ARN is ${obj.dashboardArn}")
            }
    }
}

suspend fun createDashboardWithMetrics(dashboardNameVal: String, fileNameVal:
 String) {
    val dashboardRequest = PutDashboardRequest {
        dashboardName = dashboardNameVal
        dashboardBody = readFileAsString(fileNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.putDashboard(dashboardRequest)
        println("$dashboardNameVal was successfully created.")
        val messages = response.dashboardValidationMessages
        if (messages != null) {
            if (messages.isEmpty()) {
                println("There are no messages in the new Dashboard")
            } else {
                for (message in messages) {
                    println("Message is: ${message.message}")
                }
            }
        }
    }
}

fun readFileAsString(file: String): String {
    return String(Files.readAllBytes(Paths.get(file)))
}

suspend fun getMetricStatistics(costDateWeek: String?) {
    val start = Instant.parse(costDateWeek)
    val endDate = Instant.now()
    val dimension = Dimension {
```

```
        name = "Currency"
        value = "USD"
    }

    val dimensionList: MutableList<Dimension> = ArrayList()
    dimensionList.add(dimension)

    val statisticsRequest = GetMetricStatisticsRequest {
        metricName = "EstimatedCharges"
        namespace = "AWS/Billing"
        dimensions = dimensionList
        statistics = listOf(Statistic.Maximum)
        startTime = aws.smithy.kotlin.runtime.time.Instant(start)
        endTime = aws.smithy.kotlin.runtime.time.Instant(endDate)
        period = 86400
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricStatistics(statisticsRequest)
        val data: List<Datapoint>? = response.datapoints
        if (data != null) {
            if (!data.isEmpty()) {
                for (datapoint in data) {
                    println("Timestamp:  ${datapoint.timestamp} Maximum value:
 ${datapoint.maximum}")
                }
            } else {
                println("The returned data list is empty")
            }
        }
    }
}

suspend fun getAndDisplayMetricStatistics(nameSpaceVal: String, metVal: String,
 metricOption: String, date: String, myDimension: Dimension) {
    val start = Instant.parse(date)
    val endDate = Instant.now()
    val statisticsRequest = GetMetricStatisticsRequest {
        endTime = aws.smithy.kotlin.runtime.time.Instant(endDate)
        startTime = aws.smithy.kotlin.runtime.time.Instant(start)
        dimensions = listOf(myDimension)
        metricName = metVal
        namespace = nameSpaceVal
        period = 86400
        statistics = listOf(Statistic.fromValue(metricOption))
```

```
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricStatistics(statisticsRequest)
        val data = response.datapoints
        if (data != null) {
            if (data.isNotEmpty()) {
                for (datapoint in data) {
                    println("Timestamp: ${datapoint.timestamp} Maximum value:
 ${datapoint.maximum}")
                }
            } else {
                println("The returned data list is empty")
            }
        }
    }
}

suspend fun listMets(namespaceVal: String?): ArrayList<String>? {
    val metList = ArrayList<String>()
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val reponse = cwClient.listMetrics(request)
        reponse.metrics?.forEach { metrics ->
            val data = metrics.metricName
            if (!metList.contains(data)) {
                metList.add(data!!)
            }
        }
    }
    return metList
}

suspend fun getSpecificMet(namespaceVal: String?): Dimension? {
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.listMetrics(request)
        val myList = response.metrics
        if (myList != null) {
            return myList[0].dimensions?.get(0)
```

```
        }
    }
    return null
}

suspend fun listNameSpaces(): ArrayList<String> {
    val nameSpaceList = ArrayList<String>()
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.listMetrics(ListMetricsRequest {})
        response.metrics?.forEach { metrics ->
            val data = metrics.namespace
            if (!nameSpaceList.contains(data)) {
                nameSpaceList.add(data!!)
            }
        }
    }
    return nameSpaceList
}
```

- Weitere API-Informationen finden Sie in den folgenden Themen der API-Referenz zum AWS -SDK für Kotlin.
  - DeleteAlarms
  - DeleteAnomalyDetector
  - DeleteDashboards
  - DescribeAlarmHistory
  - DescribeAlarms
  - DescribeAlarmsForMetric
  - DescribeAnomalyDetectors
  - GetMetricData
  - GetMetricStatistics
  - GetMetricWidgetImage
  - ListMetrics
  - PutAnomalyDetector
  - PutDashboard
  - PutMetricAlarm
  - PutMetricData

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

CloudWatch Metriken und Alarme mithilfe eines AWS SDK verwalten

Wie das aussehen kann, sehen Sie am nachfolgenden Beispielcode:

- Erstellen Sie einen Alarm, um eine CloudWatch Metrik zu beobachten.
- Geben Sie Daten in eine Metrik ein und lösen Sie den Alarm aus.
- Rufen Sie Daten aus dem Alarm ab.
- Löschen Sie den Alarm.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

Erstellen Sie eine Klasse, die CloudWatch Operationen umschließt.

```
from datetime import datetime, timedelta
import logging
from pprint import pprint
import random
import time
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
```

```
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def put_metric_data_set(self, namespace, name, timestamp, unit, data_set):
        """
        Sends a set of data to CloudWatch for a metric. All of the data in the
set
        have the same timestamp and unit.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param timestamp: The UTC timestamp for the metric.
        :param unit: The unit of the metric.
        :param data_set: The set of data to send. This set is a dictionary that
                         contains a list of values and a list of corresponding
counts.
                         The value and count lists must be the same length.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[
                    {
                        "MetricName": name,
                        "Timestamp": timestamp,
                        "Values": data_set["values"],
                        "Counts": data_set["counts"],
                        "Unit": unit,
                    }
                ],
            )
            logger.info("Put data set for metric %s.%s.", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data set for metric %s.%s.",
namespace, name)
            raise


    def create_metric_alarm(
        self,
```

```
        metric_namespace,
        metric_name,
        alarm_name,
        stat_type,
        period,
        eval_periods,
        threshold,
        comparison_op,
    ):
        """
        Creates an alarm that watches a metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        :param alarm_name: The name of the alarm.
        :param stat_type: The type of statistic the alarm watches.
        :param period: The period in which metric data are grouped to calculate
                       statistics.
        :param eval_periods: The number of periods that the metric must be over
the
                             alarm threshold before the alarm is set into an
alarmed
                             state.
        :param threshold: The threshold value to compare against the metric
statistic.
        :param comparison_op: The comparison operation used to compare the
threshold
                              against the metric.
        :return: The newly created alarm.
        """
        try:
            metric = self.cloudwatch_resource.Metric(metric_namespace,
metric_name)
            alarm = metric.put_alarm(
                AlarmName=alarm_name,
                Statistic=stat_type,
                Period=period,
                EvaluationPeriods=eval_periods,
                Threshold=threshold,
                ComparisonOperator=comparison_op,
            )
            logger.info(
                "Added alarm %s to track metric %s.%s.",
                alarm_name,
```

```
                metric_namespace,
                metric_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't add alarm %s to metric %s.%s",
                alarm_name,
                metric_namespace,
                metric_name,
            )
            raise
        else:
            return alarm


    def put_metric_data(self, namespace, name, value, unit):
        """
        Sends a single data value to CloudWatch for a metric. This metric is
given
        a timestamp of the current UTC time.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param value: The value of the metric.
        :param unit: The unit of the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[{"MetricName": name, "Value": value, "Unit": unit}],
            )
            logger.info("Put data for metric %s.%s", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data for metric %s.%s", namespace,
name)
            raise


    def get_metric_statistics(self, namespace, name, start, end, period,
stat_types):
        """
        Gets statistics for a metric within a specified time span. Metrics are
grouped
```

```
        into the specified period.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param start: The UTC start time of the time span to retrieve.
        :param end: The UTC end time of the time span to retrieve.
        :param period: The period, in seconds, in which to group metrics. The
period
                       must match the granularity of the metric, which depends on
                       the metric's age. For example, metrics that are older than
                       three hours have a one-minute granularity, so the period
must
                       be at least 60 and must be a multiple of 60.
        :param stat_types: The type of statistics to retrieve, such as average
value
                           or maximum value.
        :return: The retrieved statistics for the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            stats = metric.get_statistics(
                StartTime=start, EndTime=end, Period=period,
Statistics=stat_types
            )
            logger.info(
                "Got %s statistics for %s.", len(stats["Datapoints"]),
stats["Label"]
            )
        except ClientError:
            logger.exception("Couldn't get statistics for %s.%s.", namespace,
name)
            raise
        else:
            return stats


    def get_metric_alarms(self, metric_namespace, metric_name):
        """
        Gets the alarms that are currently watching the specified metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        :returns: An iterator that yields the alarms.
        """
```

```
        metric = self.cloudwatch_resource.Metric(metric_namespace, metric_name)
        alarm_iter = metric.alarms.all()
        logger.info("Got alarms for metric %s.%s.", metric_namespace,
 metric_name)
        return alarm_iter


    def delete_metric_alarms(self, metric_namespace, metric_name):
        """
        Deletes all of the alarms that are currently watching the specified
 metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(metric_namespace,
 metric_name)
            metric.alarms.delete()
            logger.info(
                "Deleted alarms for metric %s.%s.", metric_namespace, metric_name
            )
        except ClientError:
            logger.exception(
                "Couldn't delete alarms for metric %s.%s.",
                metric_namespace,
                metric_name,
            )
            raise
```

Verwenden Sie die Wrapper-Klasse, um Daten in eine Metrik einzugeben, einen Alarm auszulösen, der die Metrik überwacht, und Daten aus dem Alarm abzurufen.

```
def usage_demo():
    print("-" * 88)
    print("Welcome to the Amazon CloudWatch metrics and alarms demo!")
    print("-" * 88)

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")
```

```
    cw_wrapper = CloudWatchWrapper(boto3.resource("cloudwatch"))

    minutes = 20
    metric_namespace = "doc-example-metric"
    metric_name = "page_views"
    start = datetime.utcnow() - timedelta(minutes=minutes)
    print(
        f"Putting data into metric {metric_namespace}.{metric_name} spanning the
"
        f"last {minutes} minutes."
    )
    for offset in range(0, minutes):
        stamp = start + timedelta(minutes=offset)
        cw_wrapper.put_metric_data_set(
            metric_namespace,
            metric_name,
            stamp,
            "Count",
            {
                "values": [
                    random.randint(bound, bound * 2)
                    for bound in range(offset + 1, offset + 11)
                ],
                "counts": [random.randint(1, offset + 1) for _ in range(10)],
            },
        )

    alarm_name = "high_page_views"
    period = 60
    eval_periods = 2
    print(f"Creating alarm {alarm_name} for metric {metric_name}.")
    alarm = cw_wrapper.create_metric_alarm(
        metric_namespace,
        metric_name,
        alarm_name,
        "Maximum",
        period,
        eval_periods,
        100,
        "GreaterThanThreshold",
    )
    print(f"Alarm ARN is {alarm.alarm_arn}.")
    print(f"Current alarm state is: {alarm.state_value}.")
```

```
    print(
        f"Sending data to trigger the alarm. This requires data over the
threshold "
        f"for {eval_periods} periods of {period} seconds each."
    )
    while alarm.state_value == "INSUFFICIENT_DATA":
        print("Sending data for the metric.")
        cw_wrapper.put_metric_data(
            metric_namespace, metric_name, random.randint(100, 200), "Count"
        )
        alarm.load()
        print(f"Current alarm state is: {alarm.state_value}.")
        if alarm.state_value == "INSUFFICIENT_DATA":
            print(f"Waiting for {period} seconds...")
            time.sleep(period)
        else:
            print("Wait for a minute for eventual consistency of metric data.")
            time.sleep(period)
            if alarm.state_value == "OK":
                alarm.load()
                print(f"Current alarm state is: {alarm.state_value}.")

    print(
        f"Getting data for metric {metric_namespace}.{metric_name} during
timespan "
        f"of {start} to {datetime.utcnow()} (times are UTC)."
    )
    stats = cw_wrapper.get_metric_statistics(
        metric_namespace,
        metric_name,
        start,
        datetime.utcnow(),
        60,
        ["Average", "Minimum", "Maximum"],
    )
    print(
        f"Got {len(stats['Datapoints'])} data points for metric "
        f"{metric_namespace}.{metric_name}."
    )
    pprint(sorted(stats["Datapoints"], key=lambda x: x["Timestamp"]))

    print(f"Getting alarms for metric {metric_name}.")
    alarms = cw_wrapper.get_metric_alarms(metric_namespace, metric_name)
    for alarm in alarms:
```

```
            print(f"Alarm {alarm.name} is currently in state {alarm.state_value}.")

    print(f"Deleting alarms for metric {metric_name}.")
    cw_wrapper.delete_metric_alarms(metric_namespace, metric_name)

    print("Thanks for watching!")
    print("-" * 88)
```

- Weitere API-Informationen finden Sie in den folgenden Themen der API-Referenz zum AWS -SDK für Python (Boto3).
  - DeleteAlarms
  - DescribeAlarmsForMetric
  - DisableAlarmActions
  - EnableAlarmActions
  - GetMetricStatistics
  - ListMetrics
  - PutMetricAlarm
  - PutMetricData

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Serviceübergreifende Beispiele für die CloudWatch Verwendung von SDKs AWS

Die folgenden Beispielanwendungen verwenden AWS SDKs zur Kombination CloudWatch mit anderen. AWS-Services Jedes Beispiel enthält einen Link zu GitHub, wo Sie Anweisungen zum Einrichten und Ausführen der Anwendung finden.

Beispiele

- Überwachen Sie die Leistung von Amazon DynamoDB mithilfe eines SDK AWS

Überwachen Sie die Leistung von Amazon DynamoDB mithilfe eines SDK AWS

Das folgende Codebeispiel zeigt, wie die Verwendung von DynamoDB durch eine Anwendung zur Leistungsüberwachung konfiguriert wird.

Java

SDK für Java 2.x

Dieses Beispiel zeigt, wie eine Java-Anwendung konfiguriert wird, um die Leistung von DynamoDB zu überwachen. Die Anwendung sendet Metrikdaten an die CloudWatch Stelle, an die Sie die Leistung überwachen können.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter GitHub.

In diesem Beispiel verwendete Dienste

- CloudWatch
- DynamoDB

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwendung CloudWatch mit einem SDK AWS. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Sicherheit bei Amazon CloudWatch

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das Modell der übergreifenden Verantwortlichkeit beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Weitere Informationen zu den Compliance-Programmen, die für gelten CloudWatch, finden Sie unter AWS Services im Umfang nach Compliance-Programmen AWS .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Dienst, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung von Amazon anwenden können CloudWatch. Es zeigt Ihnen, wie Sie Amazon konfigurieren CloudWatch , um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, die Ihnen bei der Überwachung und Sicherung Ihrer CloudWatch Ressourcen helfen.

Inhalt

Datenschutz bei Amazon CloudWatch

Das AWS Modell der gilt für den Datenschutz bei Amazon CloudWatch. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle Systeme laufen AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS -Modell der geteilten Verantwortung und in der DSGVO im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie SSL/TLS, um mit Ressourcen zu kommunizieren. AWS Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein. AWS CloudTrail
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-2.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit der Konsole, der API CloudWatch oder den SDKs arbeiten oder diese anderweitig AWS-Services verwenden. AWS CLI AWS Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine

Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Verschlüsselung während der Übertragung

CloudWatch verwendet die end-to-end Verschlüsselung von Daten während der Übertragung.

Identitäts- und Zugriffsmanagement für Amazon CloudWatch

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Ressourcen zu verwenden. CloudWatch IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

- AWS verwaltete Richtlinien für Amazon CloudWatch Application Insights
- Referenz zu CloudWatch Amazon-Berechtigungen

Zielgruppe

Die Art und Weise, wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, in der Sie tätig sind. CloudWatch

Dienstbenutzer — Wenn Sie den CloudWatch Dienst für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die erforderlichen Anmeldeinformationen und Berechtigungen zur Verfügung. Wenn Sie für Ihre Arbeit mehr CloudWatch Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Fuktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anzufordern müssen. Wenn Sie in nicht auf eine Funktion zugreifen können CloudWatch, finden Sie weitere Informationen unterFehlerbehebung Amazon CloudWatch Amazon-Identität und Zugriff.

Serviceadministrator — Wenn Sie in Ihrem Unternehmen für CloudWatch Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf CloudWatch. Es ist Ihre Aufgabe, zu bestimmen, auf welche CloudWatch Funktionen und Ressourcen Ihre Servicebenutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen darüber, wie Ihr Unternehmen IAM nutzen kann CloudWatch, finden Sie unterSo CloudWatch arbeitet Amazon mit IAM.

IAM-Administrator — Wenn Sie ein IAM-Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien schreiben können, um den Zugriff darauf zu verwalten. CloudWatch Beispiele für CloudWatch identitätsbasierte Richtlinien, die Sie in IAM verwenden können, finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amazon CloudWatch

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie

sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS , übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter So melden Sie sich bei Ihrem an AWS-Konto im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, mit denen Sie Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch signieren können. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu signieren, finden Sie im IAM-Benutzerhandbuch unter AWS API-Anfragen signieren.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS Empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter Multi-Faktor-Authentifizierung im AWS IAM Identity Center - Benutzerhandbuch und Verwenden der Multi-Faktor-Authentifizierung (MFA) in AWS im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Der Zugriff erfolgt, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center - Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Eine IAM-Gruppe ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter Erstellen eines IAM-Benutzers (anstatt einer Rolle) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der übernehmen, AWS Management Console indem Sie die Rollen wechseln. Sie können eine Rolle übernehmen, indem Sie eine AWS CLI oder AWS API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter Verwenden von IAM-Rollen im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Verbundbenutzerzugriff – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter Erstellen von Rollen für externe Identitätsanbieter im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter Berechtigungssätze im AWS IAM Identity Center -Benutzerhandbuch.
- Temporäre IAM-Benutzerberechtigungen – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- Kontoübergreifender Zugriff – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.
- Serviceübergreifender Zugriff — Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon-EC2 aus oder speichert Objekte in Amazon-S3. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.

- Forward Access Sessions (FAS) — Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.
- Servicerolle – Eine Servicerolle ist eine IAM-Rolle, die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.
- Dienstbezogene Rolle — Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

- Auf Amazon EC2 ausgeführte Anwendungen — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und API-Anfragen stellen AWS CLI . AWS Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2-Instance eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instance-Profil, das an die Instance angehängt ist. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon-EC2-Instances ausgeführt werden im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter Erstellen einer IAM-Rolle (anstatt eines Benutzers) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder

Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der AWS Management Console AWS CLI, der oder der AWS API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter Auswahl zwischen verwalteten und eingebundenen Richtlinien im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Services, die ACLs unterstützen. AWS WAF Weitere Informationen“ zu ACLs finden Sie unter Zugriffskontrollliste (ACL) – Übersicht (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- Berechtigungsgrenzen – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.

- Service Control Policies (SCPs) — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen. AWS Organizations AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Objekte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Entitäten. Root-Benutzer des AWS-Kontos Weitere Informationen zu Organizations und SCPs finden Sie unter Funktionsweise von SCPs im AWS Organizations -Benutzerhandbuch.
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter Bewertungslogik für Richtlinien.

So CloudWatch arbeitet Amazon mit IAM

Bevor Sie IAM zur Verwaltung des Zugriffs auf verwenden, sollten Sie sich darüber informieren CloudWatch, mit welchen IAM-Funktionen Sie arbeiten können. CloudWatch

IAM-Funktionen, die Sie mit Amazon verwenden können CloudWatch

IAM-Feature	CloudWatch Unterstützung
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein

IAM-Feature	CloudWatch Unterstützung
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (services pezifisch)	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Teilweise
Temporäre Anmeldeinformationen	Ja
Hauptberechtigungen	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Nein

Einen allgemeinen Überblick darüber, wie CloudWatch und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren.

Identitätsbasierte Richtlinien für CloudWatch

Unterstützt Richtlinien auf Identitätsbasis.	Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer

identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für CloudWatch

Beispiele für CloudWatch identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amazon CloudWatch

Ressourcenbasierte Richtlinien finden Sie in CloudWatch

Unterstützt ressourcenbasierte Richtlinien	Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM-Administrator des vertrauenswürdigen Kontos auch der Prinzipalentität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter Wie sich IAM-Rollen von ressourcenbasierten Richtlinien unterscheiden im IAM-Benutzerhandbuch.

Richtlinienaktionen für CloudWatch

Unterstützt Richtlinienaktionen	Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der CloudWatch Aktionen finden Sie unter Von Amazon definierte Aktionen CloudWatch in der Service Authorization Reference.

Bei Richtlinienaktionen wird vor der Aktion das folgende Präfix CloudWatch verwendet:

```
cloudwatch
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
      "cloudwatch:action1",
      "cloudwatch:action2"
         ]
```

Beispiele für CloudWatch identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amazon CloudWatch

Politische Ressourcen für CloudWatch

Unterstützt Richtlinienressourcen	Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein – `Resource`oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Eine Liste der CloudWatch Ressourcentypen und ihrer ARNs finden Sie unter Von Amazon definierte Ressourcen CloudWatch in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter Von Amazon definierte Aktionen CloudWatch.

Beispiele für CloudWatch identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amazon CloudWatch

Bedingungsschlüssel für Richtlinien für CloudWatch

Unterstützt servicespezifische Richtlini enbedingungsschlüssel	Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte

Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

Eine Liste der CloudWatch Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für Amazon CloudWatch in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von Amazon definierte Aktionen CloudWatch.

Beispiele für CloudWatch identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amazon CloudWatch

ACLs in CloudWatch

Unterstützt ACLs	Nein

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit CloudWatch

Unterstützt ABAC (Tags in Richtlinien)	Teilweise

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In AWS werden diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Was ist ABAC? im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

Verwenden temporärer Anmeldeinformationen mit CloudWatch

Unterstützt temporäre Anmeldeinformationen	Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services funktionieren AWS-Services , finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter Wechseln zu einer Rolle (Konsole) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldeinformationen in IAM.

Serviceübergreifende Prinzipalberechtigungen für CloudWatch

Unterstützt Forward Access Sessions (FAS)	Ja

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für CloudWatch

Unterstützt Servicerollen	Ja

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

> **Warning**
>
> Durch das Ändern der Berechtigungen für eine Servicerolle kann die CloudWatch Funktionalität beeinträchtigt werden. Bearbeiten Sie Servicerollen nur, CloudWatch wenn Sie dazu eine Anleitung erhalten.

Beispiele für identitätsbasierte Richtlinien für Amazon CloudWatch

Standardmäßig sind Benutzer und Rollen nicht berechtigt, Ressourcen zu erstellen oder zu ändern CloudWatch. Sie können auch keine Aufgaben mithilfe der AWS Management Console, AWS Command Line Interface (AWS CLI) oder AWS API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Einzelheiten zu Aktionen und Ressourcentypen, die von definiert wurden CloudWatch, einschließlich des Formats der ARNs für jeden der Ressourcentypen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon CloudWatch in der Service Authorization Reference.

Themen

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand CloudWatch Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um Ihren Benutzern und Workloads zunächst Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter AWS -verwaltete Richtlinien oder AWS -verwaltete Richtlinien für Auftrags-Funktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer

Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.

- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung zum IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Konfigurieren eines MFA-geschützten API-Zugriffs im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der CloudWatch-Konsole

Um auf die CloudWatch Amazon-Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Informationen zu den CloudWatch Ressourcen in Ihrem Verzeichnis aufzulisten und einzusehen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen

Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen die CloudWatch Konsole weiterhin verwenden können, fügen Sie den Entitäten auch die CloudWatch `ConsoleAccess` oder die `ReadOnly` AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zu einem Benutzer im IAM-Benutzerhandbuch.

Für die CloudWatch Konsole sind Berechtigungen erforderlich

Die vollständigen Berechtigungen, die für die Arbeit mit der CloudWatch Konsole erforderlich sind, sind unten aufgeführt. Diese Berechtigungen bieten vollen Schreib- und Lesezugriff auf die CloudWatch Konsole.

- Automatische Skalierung von Anwendungen: DescribeScalingPolicies
- automatische Skalierung: DescribeAutoScalingGroups
- automatische Skalierung: DescribePolicies
- Wolkenspur: DescribeTrails
- Wolkenbeobachtung: DeleteAlarms
- Cloudwatch: DescribeAlarmHistory
- Cloudwatch: DescribeAlarms
- Cloudwatch: GetMetricData
- Cloudwatch: GetMetricStatistics
- Cloudwatch: ListMetrics
- Cloudwatch: PutMetricAlarm
- Cloudwatch: PutMetricData
- ec2: DescribeInstances
- ec2: DescribeTags
- ec2: DescribeVolumes
- ja: DescribeElasticsearchDomain
- ja: ListDomainNames

- Ereignisse: DeleteRule
- Ereignisse: DescribeRule
- Ereignisse: DisableRule
- Ereignisse: EnableRule
- Ereignisse: ListRules
- Ereignisse: PutRule
- ich bin: AttachRolePolicy
- ich bin: CreateRole
- ich bin: GetPolicy
- ich bin: GetPolicyVersion
- ich bin: GetRole
- ich bin: ListAttachedRolePolicies
- ich bin: ListRoles
- Kinese: DescribeStream
- Kinese: ListStreams
- Lambda: AddPermission
- Lambda: CreateFunction
- Lambda: GetFunctionConfiguration
- Lambda: ListAliases
- Lambda: ListFunctions
- Lambda: ListVersionsByFunction
- Lambda: RemovePermission
- Protokolle: CancelExportTask
- Protokolle: CreateExportTask
- Protokolle: CreateLogGroup
- Protokolle: CreateLogStream
- Protokolle: DeleteLogGroup
- Protokolle: DeleteLogStream
- Protokolle: DeleteMetricFilter

- Protokolle: DeleteRetentionPolicy
- Protokolle: DeleteSubscriptionFilter
- Protokolle: DescribeExportTasks
- Protokolle: DescribeLogGroups
- Protokolle: DescribeLogStreams
- Protokolle: DescribeMetricFilters
- Protokolle: DescribeQueries
- Protokolle: DescribeSubscriptionFilters
- Protokolle: FilterLogEvents
- Protokolle: GetLogGroupFields
- Protokolle: GetLogRecord
- Protokolle: GetLogEvents
- Protokolle: GetQueryResults
- Protokolle: PutMetricFilter
- Protokolle: PutRetentionPolicy
- Protokolle: PutSubscriptionFilter
- Protokolle: StartQuery
- Protokolle: StopQuery
- Protokolle: TestMetricFilter
- s3: CreateBucket
- s3: ListBucket
- sns: CreateTopic
- sns: GetTopicAttributes
- sns: ListSubscriptions
- sns: ListTopics
- sns: SetTopicAttributes
- sns:Subscribe
- sns:Unsubscribe
- sqs: GetQueueAttributes

- sqs: GetQueueUrl
- sqs: ListQueues
- sqs: SetQueueAttributes
- swf: CreateAction
- swf: DescribeAction
- swf: ListActionTemplates
- swf: RegisterAction
- swf: RegisterDomain
- swf: UpdateAction

Um die X-Ray Trace Map anzuzeigen, benötigen Sie außerdem `AWSXrayReadOnlyAccess`

Fehlerbehebung Amazon CloudWatch Amazon-Identität und Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit CloudWatch IAM auftreten können.

Themen

- Ich bin nicht berechtigt, eine Aktion durchzuführen in CloudWatch
- Ich bin nicht berechtigt, iam auszuführen: PassRole
- Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine CloudWatch Ressourcen ermöglichen

Ich bin nicht berechtigt, eine Aktion durchzuführen in CloudWatch

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven *`my-example-widget`*-Ressource anzuzeigen, jedoch nicht über `cloudwatch:`*`GetWidget`*-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 cloudwatch:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `cloudwatch:GetWidget`-Aktion auf die *`my-example-widget`*-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht berechtigt sind, die `iam:PassRole` Aktion auszuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an CloudWatch diese Person übergeben können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in auszuführen. CloudWatch Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine CloudWatch Ressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob diese Funktionen CloudWatch unterstützt werden, finden Sie unterSo CloudWatch arbeitet Amazon mit IAM.
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.

CloudWatch Aktualisierung der Dashboard-Berechtigungen

Am 1. Mai 2018 wurden die für den Zugriff auf CloudWatch Dashboards erforderlichen Berechtigungen AWS geändert. Für den Dashboard-Zugriff in der CloudWatch Konsole sind jetzt Berechtigungen erforderlich, die 2017 eingeführt wurden, um Dashboard-API-Operationen zu unterstützen:

- Cloudwatch: GetDashboard
- Cloudwatch: ListDashboards
- Cloudwatch: PutDashboard
- Cloudwatch: DeleteDashboards

Um auf CloudWatch Dashboards zugreifen zu können, benötigen Sie eine der folgenden Voraussetzungen:

- Die AdministratorAccessRichtlinie.
- Die CloudWatchFullAccessRichtlinie.
- Eine benutzerdefinierte Richtlinie mit einem oder mehreren dieser spezifischen Berechtigungen:
  - `cloudwatch:GetDashboard` und `cloudwatch:ListDashboards`, um Dashboards anzeigen zu können

- `cloudwatch:PutDashboard`, um Dashboards erstellen oder ändern zu können
- `cloudwatch:DeleteDashboards`, um Dashboards löschen zu können

Weitere Informationen zur Verwendung von Richtlinien zum Ändern von Berechtigungen für einen IAM-Benutzer finden Sie unter Ändern von Berechtigungen für einen IAM-Benutzer.

Weitere Informationen zu CloudWatch Berechtigungen finden Sie unterReferenz zu CloudWatch Amazon-Berechtigungen.

Weitere Informationen zu Dashboard-API-Vorgängen finden Sie PutDashboardin der Amazon CloudWatch API-Referenz.

AWS verwaltete (vordefinierte) Richtlinien für CloudWatch

AWS adressiert viele gängige Anwendungsfälle durch die Bereitstellung eigenständiger IAM-Richtlinien, die von erstellt und verwaltet AWS werden. Diese AWS verwalteten Richtlinien gewähren die erforderlichen Berechtigungen für allgemeine Anwendungsfälle, sodass Sie nicht erst untersuchen müssen, welche Berechtigungen benötigt werden. Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

Die folgenden AWS verwalteten Richtlinien, die Sie Benutzern in Ihrem Konto zuordnen können, sind spezifisch für CloudWatch.

Themen

- CloudWatchFullAccessV2
- CloudWatchFullAccess
- CloudWatchReadOnlyAccess
- CloudWatchActionsEC2-Zugriff
- CloudWatchAutomaticDashboardsAccess
- CloudWatchAgentServerPolicy
- CloudWatchAgentAdminPolicy
- AWS verwaltete (vordefinierte) Richtlinien für CloudWatch kontoübergreifende Beobachtbarkeit
- AWS verwaltete (vordefinierte) Richtlinien für CloudWatch Synthetics
- AWS verwaltete (vordefinierte) Richtlinien für Amazon CloudWatch RUM
- AWS verwaltete (vordefinierte) Richtlinien für CloudWatch Evidently
- AWS verwaltete Richtlinie für AWS Systems Manager Incident Manager

CloudWatchFullAccessV2

AWS hat kürzlich die verwaltete CloudWatchFullAccessV2-IAM-Richtlinie hinzugefügt. Diese Richtlinie gewährt vollen Zugriff auf CloudWatch Aktionen und Ressourcen und legt auch den Geltungsbereich der für andere Dienste wie Amazon SNS und erteilten Berechtigungen genauer fest. Amazon EC2 Auto Scaling Wir empfehlen, dass Sie mit der Verwendung dieser Richtlinie beginnen, anstatt sie zu verwenden. CloudWatchFullAccess AWS plant, CloudWatchFullAccessin naher future nicht mehr zu unterstützen.

Es enthält `application-signals:` Berechtigungen, sodass Benutzer von der CloudWatch Konsole aus unter Application Signals auf alle Funktionen zugreifen können. Es enthält einige `autoscaling:Describe` Berechtigungen, sodass Benutzer mit dieser Richtlinie die Auto Scaling Scaling-Aktionen sehen können, die mit CloudWatch Alarmen verknüpft sind. Es enthält einige `sns` Berechtigungen, sodass Benutzer mit dieser Richtlinie Amazon SNS SNS-Themen abrufen und sie mit CloudWatch Alarmen verknüpfen können. Sie umfasst IAM-Berechtigungen, sodass Benutzer mit dieser Richtlinie Informationen über die Rollen im Zusammenhang mit Services einsehen können, mit denen verknüpft ist. CloudWatch Sie umfasst die `oam:ListAttachedLinks` Berechtigungen `oam:ListSinks` und, sodass Benutzer mit dieser Richtlinie die Konsole verwenden können, um gemeinsam genutzte Daten von Quellkonten CloudWatch kontenübergreifend einzusehen.

Es umfasst `rumsynthetics`, und `xray` Berechtigungen, sodass Benutzer vollen Zugriff auf CloudWatch Synthetics, und CloudWatch RUM haben können AWS X-Ray, die alle im Rahmen des CloudWatch Dienstes enthalten sind.

Der Inhalt von CloudWatchFullAccessV2 lautet wie folgt:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchFullAccessPermissions",
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DescribeScalingPolicies",
                "application-signals:*",
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribePolicies",
                "cloudwatch:*",
                "logs:*",
                "sns:CreateTopic",
                "sns:ListSubscriptions",
```

```
                "sns:ListSubscriptionsByTopic",
                "sns:ListTopics",
                "sns:Subscribe",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "oam:ListSinks",
                "rum:*",
                "synthetics:*",
                "xray:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsServiceLinkedRolePermissions",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "application-
signals.cloudwatch.amazonaws.com"
                }
            }
        },
        {
            "Sid": "EventsServicePermissions",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/events.amazonaws.com/
AWSServiceRoleForCloudWatchEvents*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "events.amazonaws.com"
                }
            }
        },
         {
            "Sid": "OAMReadPermissions",
            "Effect": "Allow",
            "Action": [
                "oam:ListAttachedLinks"
            ],
```

```
            "Resource": "arn:aws:oam:*:*:sink/*"
        }
    ]
}
```

CloudWatchFullAccess

Die CloudWatchFullAccessRichtlinie ist auf dem Weg, veraltet zu sein. Wir empfehlen, dass Sie sie nicht mehr verwenden und stattdessen CloudWatchFullAccessV2 verwenden.

Der Inhalt von CloudWatchFullAccesslautet wie folgt:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:Describe*",
                "cloudwatch:*",
                "logs:*",
                "sns:*",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "oam:ListSinks"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/events.amazonaws.com/
AWSServiceRoleForCloudWatchEvents*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "events.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "oam:ListAttachedLinks"
            ],
            "Resource": "arn:aws:oam:*:*:sink/*"
        }
    ]
}
```

CloudWatchReadOnlyAccess

Die CloudWatchReadOnlyAccessRichtlinie gewährt nur Lesezugriff auf. CloudWatch

Die Richtlinie beinhaltet einige `logs:` Berechtigungen, sodass Benutzer mit dieser Richtlinie die Konsole verwenden können, um CloudWatch Logs-Informationen und CloudWatch Logs Insights-Abfragen anzuzeigen. Es beinhaltet`autoscaling:Describe*`, sodass Benutzer mit dieser Richtlinie die Auto Scaling Scaling-Aktionen sehen können, die mit CloudWatch Alarmen verknüpft sind. Es umfasst die `application-signals:` Berechtigungen, mit denen Benutzer Application Signals verwenden können, um den Zustand ihrer Dienste zu überwachen. `application-autoscaling:DescribeScalingPolicies` ist enthalten, damit Benutzer mit dieser Richtlinie auf Informationen über Richtlinien für Application Auto Scaling zugreifen können. Es beinhaltet `sns:Get*` und`sns:List*`, sodass Benutzer mit dieser Richtlinie Informationen zu den Amazon SNS SNS-Themen abrufen können, die Benachrichtigungen über CloudWatch Alarme erhalten. Es umfasst die `oam:ListAttachedLinks` Berechtigungen `oam:ListSinks` und, sodass Benutzer mit dieser Richtlinie die Konsole verwenden können, um Daten, die von Quellkonten geteilt wurden, CloudWatch kontenübergreifend einzusehen. Sie enthält die `iam:GetRole` Berechtigungen, mit denen Benutzer überprüfen können, ob CloudWatch Application Signals eingerichtet wurden.

Es umfasst `rumsynthetics`, und `xray` Berechtigungen, sodass Benutzer nur Lesezugriff auf CloudWatch Synthetics und CloudWatch RUM haben können AWS X-Ray, die alle im Rahmen des Dienstes enthalten sind. CloudWatch

Im Folgenden finden Sie den Inhalt der Richtlinie. CloudWatchReadOnlyAccess

```
{
"Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchReadOnlyAccessPermissions",
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DescribeScalingPolicies",
                "application-signals:BatchGet*",
```

```
                "application-signals:Get*",
                "application-signals:List*",
                "autoscaling:Describe*",
                "cloudwatch:BatchGet*",
                "cloudwatch:Describe*",
                "cloudwatch:GenerateQuery",
                "cloudwatch:Get*",
                "cloudwatch:List*",
                "logs:Get*",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:Describe*",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents",
                "logs:StartLiveTail",
                "logs:StopLiveTail",
                "oam:ListSinks",
                "sns:Get*",
                "sns:List*",
                "rum:BatchGet*",
                "rum:Get*",
                "rum:List*",
                "synthetics:Describe*",
                "synthetics:Get*",
                "synthetics:List*",
                "xray:BatchGet*",
                "xray:Get*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "OAMReadPermissions",
            "Effect": "Allow",
            "Action": [
                "oam:ListAttachedLinks"
            ],
            "Resource": "arn:aws:oam:*:*:sink/*"
        },
        {
            "Sid": "CloudWatchReadOnlyGetRolePermissions",
            "Effect": "Allow",
            "Action": "iam:GetRole",
```

```
            "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals"
        }
    ]
}
```

CloudWatchActionsEC2-Zugriff

Die CloudWatchActionsEC2Access-Richtlinie gewährt zusätzlich zu den Amazon EC2-Metadaten nur Lesezugriff auf CloudWatch Alarme und Metriken. Gewährt Zugriff zum Anhalten, Beenden und Neustarten von API-Aktionen für EC2-Instances.

Im Folgenden finden Sie den Inhalt der EC2Access-Richtlinie. CloudWatchActions

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:Describe*",
        "ec2:Describe*",
        "ec2:RebootInstances",
        "ec2:StopInstances",
        "ec2:TerminateInstances"
      ],
      "Resource": "*"
    }
  ]
}
```

CloudWatchAutomaticDashboardsAccess

Die CloudWatchCrossAccountAccessverwaltete Richtlinie wird von der CloudWatch CrossAccountSharingRole IAM-Rolle verwendet. Mit dieser Rolle und Richtlinie können Benutzer von kontoübergreifenden Dashboards in jedem Konto, das Dashboards freigibt, automatische Dashboards anzeigen.

Im Folgenden finden Sie den Inhalt von CloudWatchAutomaticDashboardsAccess:

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Action": [
        "autoscaling:DescribeAutoScalingGroups",
        "cloudfront:GetDistribution",
        "cloudfront:ListDistributions",
        "dynamodb:DescribeTable",
        "dynamodb:ListTables",
        "ec2:DescribeInstances",
        "ec2:DescribeVolumes",
        "ecs:DescribeClusters",
        "ecs:DescribeContainerInstances",
        "ecs:ListClusters",
        "ecs:ListContainerInstances",
        "ecs:ListServices",
        "elasticache:DescribeCacheClusters",
        "elasticbeanstalk:DescribeEnvironments",
        "elasticfilesystem:DescribeFileSystems",
        "elasticloadbalancing:DescribeLoadBalancers",
        "kinesis:DescribeStream",
        "kinesis:ListStreams",
        "lambda:GetFunction",
        "lambda:ListFunctions",
        "rds:DescribeDBClusters",
        "rds:DescribeDBInstances",
        "resource-groups:ListGroupResources",
        "resource-groups:ListGroups",
        "route53:GetHealthCheck",
        "route53:ListHealthChecks",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "sns:ListTopics",
        "sqs:GetQueueAttributes",
        "sqs:GetQueueUrl",
        "sqs:ListQueues",
        "synthetics:DescribeCanariesLastRun",
        "tag:GetResources"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "apigateway:GET"
```

```
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:apigateway:*::/restapis*"
      ]
    }
  ]
```

CloudWatchAgentServerPolicy

Die CloudWatchAgentServerPolicyRichtlinie kann in IAM-Rollen verwendet werden, die Amazon EC2 EC2-Instances zugeordnet sind, damit der CloudWatch Agent Informationen aus der Instance lesen und in sie schreiben kann. CloudWatch Ihr Inhalt lautet wie folgt.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CWACloudWatchServerPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "ec2:DescribeVolumes",
                "ec2:DescribeTags",
                "logs:PutLogEvents",
                "logs:PutRetentionPolicy",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "xray:PutTraceSegments",
                "xray:PutTelemetryRecords",
                "xray:GetSamplingRules",
                "xray:GetSamplingTargets",
                "xray:GetSamplingStatisticSummaries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CWASSMServerPermissions",
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter"
```

```
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/AmazonCloudWatch-*"
        }
    ]
}
```

CloudWatchAgentAdminPolicy

Die CloudWatchAgentAdminPolicyRichtlinie kann in IAM-Rollen verwendet werden, die Amazon EC2 EC2-Instances zugeordnet sind. Diese Richtlinie ermöglicht es dem CloudWatch Agenten, Informationen aus der Instance zu lesen und in sie zu CloudWatch schreiben sowie Informationen in den Parameter Store zu schreiben. Ihr Inhalt lautet wie folgt.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CWACloudWatchPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "ec2:DescribeTags",
                "logs:PutLogEvents",
                "logs:PutRetentionPolicy",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "xray:PutTraceSegments",
                "xray:PutTelemetryRecords",
                "xray:GetSamplingRules",
                "xray:GetSamplingTargets",
                "xray:GetSamplingStatisticSummaries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CWASSMPermissions",
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter",
                "ssm:PutParameter"
            ],
```

```
            "Resource": "arn:aws:ssm:*:*:parameter/AmazonCloudWatch-*"
        }
    ]
}
```

> **Note**
>
> Sie können diese Berechtigungsrichtlinien prüfen, indem Sie sich bei der IAM-Konsole anmelden und dort nach bestimmten Richtlinien suchen.

Sie können auch Ihre eigenen benutzerdefinierten IAM-Richtlinien erstellen, um Berechtigungen für CloudWatch Aktionen und Ressourcen zu gewähren. Die benutzerdefinierten Richtlinien können Sie dann den IAM-Benutzern oder -Gruppen zuweisen, die diese Berechtigungen benötigen.

AWS verwaltete (vordefinierte) Richtlinien für CloudWatch kontoübergreifende Beobachtbarkeit

Die Richtlinien in diesem Abschnitt gewähren Berechtigungen im Zusammenhang mit der CloudWatch kontoübergreifenden Beobachtbarkeit. Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.

CloudWatchCrossAccountSharingConfiguration

Die CloudWatchCrossAccountSharingConfigurationRichtlinie gewährt Zugriff auf das Erstellen, Verwalten und Anzeigen von Observability Access Manager-Links für die gemeinsame Nutzung von CloudWatch Ressourcen zwischen Konten. Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit. Der Inhalt ist wie folgt:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:Link",
                "oam:ListLinks"
            ],
            "Resource": "*"
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
            ],
            "Resource": "arn:aws:oam:*:*:link/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Resource": [
                "arn:aws:oam:*:*:link/*",
                "arn:aws:oam:*:*:sink/*"
            ]
        }
    ]
}
```

OAM FullAccess

Die FullAccessOAM-Richtlinie gewährt Zugriff auf die Erstellung, Verwaltung und Anzeige von Observability Access Manager-Senken und -Links, die für kontoübergreifende Observability verwendet werden. CloudWatch

Die FullAccessOAM-Richtlinie allein erlaubt es Ihnen nicht, Observability-Daten linkübergreifend auszutauschen. Um einen Link zum Teilen von CloudWatch Metriken zu erstellen, benötigen Sie außerdem entweder oder CloudWatchFullAccess. CloudWatchCrossAccountSharingConfiguration Um einen Link zum Teilen von CloudWatch Logs-Protokollgruppen zu erstellen, benötigen Sie außerdem entweder CloudWatchLogsFullAccessoder CloudWatchLogsCrossAccountSharingConfiguration. Um einen Link zum Teilen von X-Ray-Traces zu erstellen, benötigen Sie außerdem entweder AWSXRayFullAccessoder AWSXRayCrossAccountSharingConfiguration.

Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit. Der Inhalt ist wie folgt:

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "oam:*"
            ],
            "Resource": "*"
        }
    ]
}
```

OAM ReadOnlyAccess

Die ReadOnlyAccessOAM-Richtlinie gewährt schreibgeschützten Zugriff auf Observability Access Manager-Ressourcen, die für kontoübergreifende Observability verwendet werden. CloudWatch Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit. Der Inhalt ist wie folgt:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "oam:Get*",
                "oam:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS verwaltete (vordefinierte) Richtlinien für CloudWatch Synthetics

Die CloudWatchSyntheticsFullAccessund CloudWatchSyntheticsReadOnlyAccess AWS verwalteten Richtlinien stehen Ihnen zur Verfügung, um sie Benutzern zuzuweisen, die CloudWatch Synthetics verwalten oder verwenden. Die folgenden zusätzlichen Richtlinien sind ebenfalls relevant:

- AmazonS3 ReadOnlyAccess und CloudWatchReadOnlyAccess— Diese sind notwendig, um alle Synthetics-Daten in der Konsole lesen zu können. CloudWatch

- AWSLambdaReadOnlyAccess— Um den von Canaries verwendeten Quellcode einsehen zu können.
- CloudWatchSyntheticsFullAccessermöglicht es Ihnen, Canaries zu erstellen. Um Canaries zu erstellen und zu löschen, für die eine neue IAM-Rolle erstellt wurde, benötigen Sie zusätzlich die folgende Inline-Richtlinienanweisung:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:CreatePolicy",
                "iam:DeletePolicy",
                "iam:AttachRolePolicy",
                "iam:DetachRolePolicy",
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*",
                "arn:aws:iam::*:policy/service-role/CloudWatchSyntheticsPolicy*"
            ]
        }
    ]
}
```

Important

Wenn Sie einem Benutzer die Berechtigungen `iam:CreateRole`, `iam:DeleteRole`, `iam:CreatePolicy`, `iam:DeletePolicy`, `iam:AttachRolePolicy` und `iam:DetachRolePolicy` gewähren, erhält dieser Benutzer vollen Administratorzugriff zum Erstellen, Anfügen und Löschen von Rollen und Richtlinien mit ARNs, die mit `arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*` und `arn:aws:iam::*:policy/service-role/CloudWatchSyntheticsPolicy*` übereinstimmen. Beispielsweise kann ein Benutzer mit diesen Berechtigungen eine Richtlinie erstellen, die über vollständige Berechtigungen für alle Ressourcen verfügt, und diese Richtlinie an eine beliebige Rolle anhängen, die mit diesem ARN-Muster übereinstimmt. Seien Sie sehr vorsichtig, wem Sie diese Berechtigungen erteilen.

Informationen zum Anhängen von Richtlinien und zum Erteilen von Berechtigungen für Benutzer finden Sie unter Ändern von Berechtigungen für einen IAM-Benutzer und So betten Sie eine Inline-Richtlinie für einen Benutzer oder eine Rolle ein.

CloudWatchSyntheticsFullAccess

Im Folgenden finden Sie den Inhalt der Richtlinie. CloudWatchSyntheticsFullAccess

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutEncryptionConfiguration"
            ],
            "Resource": [
                "arn:aws:s3:::cw-syn-results-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles",
                "s3:ListAllMyBuckets",
                "xray:GetTraceSummaries",
                "xray:BatchGetTraces",
                "apigateway:GET"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::cw-syn-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::aws-synthetics-library-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "lambda.amazonaws.com",
                        "synthetics.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:ListAttachedRolePolicies"
            ],
            "Resource": [
```

```
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DeleteAlarms"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:Synthetics-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:CreateFunction",
                "lambda:AddPermission",
                "lambda:PublishVersion",
                "lambda:UpdateFunctionCode",
                "lambda:UpdateFunctionConfiguration",
                "lambda:GetFunctionConfiguration",
                "lambda:DeleteFunction"
            ],
            "Resource": [
                "arn:aws:lambda:*:*:function:cwsyn-*"
            ]
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:GetLayerVersion",
                "lambda:PublishLayerVersion",
                "lambda:DeleteLayerVersion"
            ],
            "Resource": [
                "arn:aws:lambda:*:*:layer:cwsyn-*",
                "arn:aws:lambda:*:*:layer:Synthetics:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:ListTopics"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:ListSubscriptionsByTopic"
            ],
            "Resource": [
                "arn:*:sns:*:*:Synthetics-*"
            ]
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "kms:ListAliases"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:*:*:key/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:*:*:key/*",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "s3.*.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

CloudWatchSyntheticsReadOnlyAccess

Das Folgende ist der Inhalt der CloudWatchSyntheticsReadOnlyAccessRichtlinie.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:Describe*",
                "synthetics:Get*",
                "synthetics:List*",
```

```
                "lambda:GetFunctionConfiguration"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS verwaltete (vordefinierte) Richtlinien für Amazon CloudWatch RUM

Die ReadOnlyAccess AWS verwalteten AmazonCloudWatchAmazonCloudWatchRUM FullAccess - und RUM-Richtlinien können Sie Benutzern zuweisen, die CloudWatch RUM verwalten oder verwenden.

AmazonCloudWatchRUM FullAccess

Im Folgenden sind die Inhalte der AmazonCloudWatchFullAccessRUM-Richtlinie aufgeführt.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "rum:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/aws-service-role/rum.amazonaws.com/
AWSServiceRoleForRealUserMonitoring"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
```

```
            "Resource": [
                "arn:aws:iam::*:role/RUM-Monitor*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "cognito-identity.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms"
            ],
            "Resource": "arn:aws:cloudwatch:*:*:alarm:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cognito-identity:CreateIdentityPool",
                "cognito-identity:ListIdentityPools",
                "cognito-identity:DescribeIdentityPool",
                "cognito-identity:GetIdentityPoolRoles",
                "cognito-identity:SetIdentityPoolRoles"
            ],
            "Resource": "arn:aws:cognito-identity:*:*:identitypool/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:DeleteLogGroup",
                "logs:PutRetentionPolicy",
```

```
                "logs:CreateLogStream"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:*RUMService*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries",
                "logs:DescribeResourcePolicies"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": "arn:aws:logs:*:*:log-group::log-stream:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:describeCanaries",
                "synthetics:describeCanariesLastRun"
            ],
            "Resource": "arn:aws:synthetics:*:*:canary:*"
        }
    ]
}
```

AmazonCloudWatchRUM ReadOnlyAccess

Im Folgenden sind die Inhalte der AmazonCloudWatchReadOnlyAccessRUM-Richtlinie aufgeführt.

```
{

 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
```

```
    "Action": [
     "rum:GetAppMonitor",
     "rum:GetAppMonitorData",
     "rum:ListAppMonitors",
     "rum:ListRumMetricsDestinations",
     "rum:BatchGetRumMetricDefinitions"
    ],
    "Resource": "*"
   }
  ]
}
```

AmazonCloudWatchRUM ServiceRolePolicy

Sie können AmazonCloudWatchRUM nicht ServiceRolePolicy an Ihre IAM-Entitäten anhängen. Diese Richtlinie ist mit einer dienstbezogenen Rolle verknüpft, die es CloudWatch RUM ermöglicht, Überwachungsdaten für andere relevante AWS Dienste zu veröffentlichen. Weitere Informationen zu dieser serviceverknüpften Rolle finden Sie unter Verwendung von serviceverknüpften Rollen für RUM CloudWatch.

Der vollständige Inhalt von AmazonCloudWatchRUM ServiceRolePolicy lautet wie folgt.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "xray:PutTraceSegments"
   ],
   "Resource": [
    "*"
   ]
  },
  {
   "Effect": "Allow",
   "Action": "cloudwatch:PutMetricData",
   "Resource": "*",
   "Condition": {
    "StringLike": {
     "cloudwatch:namespace": [
      "RUM/CustomMetrics/*",
      "AWS/RUM"
```

```
      ]
     }
    }
   }
  ]
}
```

AWS verwaltete (vordefinierte) Richtlinien für CloudWatch Evidently

Die CloudWatchEvidentlyFullAccessund CloudWatchEvidentlyReadOnlyAccess AWS verwalteten Richtlinien stehen Ihnen zur Verfügung, damit Sie sie Benutzern zuweisen können, die CloudWatch Evidently verwalten oder verwenden werden.

CloudWatchEvidentlyFullAccess

Im Folgenden finden Sie den Inhalt der CloudWatchEvidentlyFullAccessRichtlinie.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "evidently:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchRUMEvidentlyRole-*"
            ]
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:DescribeAlarmHistory",
                "cloudwatch:DescribeAlarmsForMetric",
                "cloudwatch:ListTagsForResource"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms",
                "cloudwatch:TagResource",
                "cloudwatch:UnTagResource"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudtrail:LookupEvents"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricAlarm"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:Evidently-Alarm-*"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:ListTopics"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:ListSubscriptionsByTopic"
            ],
            "Resource": [
                "arn:*:sns:*:*:Evidently-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

CloudWatchEvidentlyReadOnlyAccess

Im Folgenden finden Sie den Inhalt der CloudWatchEvidentlyReadOnlyAccessRichtlinie.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "evidently:GetExperiment",
                "evidently:GetFeature",
                "evidently:GetLaunch",
                "evidently:GetProject",
                "evidently:GetSegment",
                "evidently:ListExperiments",
                "evidently:ListFeatures",
                "evidently:ListLaunches",
                "evidently:ListProjects",
                "evidently:ListSegments",
                "evidently:ListSegmentReferencs"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS verwaltete Richtlinie für AWS Systems Manager Incident Manager

Die AWSCloudWatchAlarms_ActionSSMIncidentsServiceRolePolicyRichtlinie ist einer dienstbezogenen Rolle zugeordnet, die es ermöglicht, in Ihrem Namen Incidents in AWS Systems Manager Incident Manager CloudWatch zu starten. Weitere Informationen finden Sie unter Dienstbezogene Rollenberechtigungen für CloudWatch Alarme Systems Manager Incident Manager-Aktionen.

Die Richtlinie hat die folgende Berechtigung:

- SSM-Vorfälle: StartIncident

Beispiele für vom Kunden verwaltete Richtlinien

In diesem Abschnitt finden Sie Beispiele für Benutzerrichtlinien, die Berechtigungen für verschiedene CloudWatch Aktionen gewähren. Diese Richtlinien funktionieren, wenn Sie die CloudWatch API, AWS SDKs oder die AWS CLI verwenden.

Beispiele

- Beispiel 1: Erlauben Sie dem Benutzer vollen Zugriff auf CloudWatch
- Beispiel 2: Erlauben Sie den schreibgeschützten Zugriff auf CloudWatch
- Beispiel 3: Anhalten oder Beenden einer Amazon-EC2-Instance

Beispiel 1: Erlauben Sie dem Benutzer vollen Zugriff auf CloudWatch

Um einem Benutzer vollen Zugriff auf zu gewähren CloudWatch, können Sie „Gewähren Sie ihm die CloudWatchFullAccessverwaltete Richtlinie“ verwenden, anstatt eine vom Kunden verwaltete Richtlinie zu erstellen. Der Inhalt von ist CloudWatchFullAccessunter aufgeführt. CloudWatchFullAccess

Beispiel 2: Erlauben Sie den schreibgeschützten Zugriff auf CloudWatch

Die folgende Richtlinie ermöglicht Benutzern den schreibgeschützten Zugriff auf Amazon EC2 Auto Scaling Scaling-Aktionen, CloudWatch -Metriken, CloudWatch Protokolldaten und alarmbezogene Amazon SNS SNS-Daten. CloudWatch

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "autoscaling:Describe*",
        "cloudwatch:Describe*",
        "cloudwatch:Get*",
        "cloudwatch:List*",
        "logs:Get*",
        "logs:Describe*",
        "logs:StartQuery",
        "logs:StopQuery",
        "logs:TestMetricFilter",
        "logs:FilterLogEvents",
        "logs:StartLiveTail",
        "logs:StopLiveTail",
        "sns:Get*",
        "sns:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Beispiel 3: Anhalten oder Beenden einer Amazon-EC2-Instance

Die folgende Richtlinie ermöglicht eine CloudWatch Alarmaktion zum Stoppen oder Beenden einer EC2-Instance. Im folgenden Beispiel sind die DescribeAlarms Aktionen GetMetricData ListMetrics, und optional. Es wird empfohlen, dass Sie diese Aktionen einfügen, um sicherzustellen, dass Sie die Instance ordnungsgemäß angehalten oder beendet haben.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:GetMetricData",
        "cloudwatch:ListMetrics",
        "cloudwatch:DescribeAlarms"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstances",
        "ec2:StopInstances",
        "ec2:TerminateInstances"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

CloudWatch Aktualisierungen der AWS verwalteten Richtlinien

Hier finden Sie Informationen zu Aktualisierungen AWS verwalteter Richtlinien, die CloudWatch seit Beginn der Nachverfolgung dieser Änderungen durch diesen Dienst vorgenommen wurden.

Abonnieren Sie den RSS-Feed auf der Seite CloudWatch Dokumentenverlauf, um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten.

Änderung	Beschreibung	Datum
CloudWatchFullAccessV2 — Aktualisierung einer bestehenden Richtlinie	CloudWatch hat die Richtlinie mit dem Namen CloudWatchFullAccessV2 aktualisiert. Der Geltungsbereich der `CloudWatchFullAccessPermissions` Richtlinie wurde dahingehend aktualisiert, dass Benutzer nun CloudWatch Application Signals verwenden können, um Probleme mit dem Zustand ihrer Dienste einzusehen, zu untersuchen und zu diagnostizieren. `application-signals:*`	20. Mai 2024
CloudWatchReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	CloudWatch hat die genannte Richtlinie aktualisiert CloudWatchReadOnlyAccess. Der Geltungsbereich der `CloudWatchReadOnlyAccessPermissions` Richtlinie wurde dahingehend aktualisiert `application-signals:BatchGet* application-signals:List*` , dass Benutzer CloudWatch Anwendungssignale verwenden können, um Probleme mit dem Zustand	20. Mai 2024

Änderung	Beschreibung	Datum
	ihrer Dienste einzusehen, zu untersuchen und zu diagnostizieren. `application-signals:Get*` Der Geltungsbereich von `CloudWatchReadOnlyGetRolePermissions` wurde um die `iam:GetRole` Aktion erweitert, sodass Benutzer überprüfen können, ob CloudWatch Application Signals eingerichtet ist.	
CloudWatchApplicationSignalsServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	CloudWatch hat die genannte Richtlinie aktualisiert CloudWatchApplicationSignalsServiceRolePolicy. Der Geltungsbereich der `logs:GetQueryResults` Berechtigungen `logs:StartQuery` und wurde dahingehend geändert, dass ARNs `arn:aws:logs:*:*:log-group:/aws/appsignals/*:*` und `arn:aws:logs:*:*:log-group:/aws/application-signals/data:*` ARNs hinzugefügt wurden, um Application Signals auf mehr Architekturen zu aktivieren.	18. April 2024

Änderung	Beschreibung	Datum
CloudWatchApplicationSignalsServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	CloudWatch hat den Umfang einer Genehmigung in geändert CloudWatchApplicationSignalsServiceRolePolicy. Der Umfang der `cloudwatch:GetMetricData` Genehmigung wurde dahingehend geändert, * dass Application Signals Messwerte aus Quellen verknüpfter Konten abrufen kann.	08. April 2024
CloudWatchAgentServerPolicy – Aktualisierung auf eine bestehende Richtlinie	CloudWatch hat Berechtigungen zu hinzugefügt CloudWatchAgentServerPolicy. Die `logs:PutRetentionPolicy` Berechtigungen `xray:PutTraceSegments` `xray:PutTelemetryRecords` , `xray:GetSamplingRules` `xray:GetSamplingTargets` , `xray:GetSamplingStatisticSummaries` und wurden hinzugefügt, sodass der CloudWatch Agent X-Ray-Traces veröffentlichen und die Aufbewahrungsfristen für Protokollgruppen ändern kann.	12. Februar 2024

Änderung	Beschreibung	Datum
CloudWatchAgentAdminPolicy – Aktualisierung auf eine bestehende Richtlinie	CloudWatch hat Berechtigungen zu hinzugefügt CloudWatchAgentAdminPolicy. Die `logs:PutRetentionPolicy` Berechtigungen `xray:PutTraceSegments` `xray:PutTelemetryRecords` , `xray:GetSamplingRules` `xray:GetSamplingTargets` , `xray:GetSamplingStatisticSummaries` und wurden hinzugefügt, sodass der CloudWatch Agent X-Ray-Traces veröffentlichen und die Aufbewahrungsfristen für Protokollgruppen ändern kann.	12. Februar 2024

Änderung	Beschreibung	Datum
CloudWatchFullAccessV2 — Aktualisierung einer bestehenden Richtlinie	CloudWatch Berechtigungen zu CloudWatchFullAccessV2 hinzugefügt. Bestehende Berechtigungen für CloudWatch Synthetics-, X-Ray- und CloudWatch RUM-Aktionen sowie neue Berechtigungen für CloudWatch Application Signals wurden hinzugefügt, sodass Benutzer mit dieser Richtlinie CloudWatch Application Signals verwalten können. Die Berechtigung zum Erstellen der dienstbezogenen Rolle „ CloudWatch Application Signals“ wurde hinzugefügt, damit CloudWatch Application Signals Telemetriedaten in Logs, Metriken, Traces und Tags ermitteln kann.	05. Dezember 2023

Änderung	Beschreibung	Datum
CloudWatchReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	CloudWatch Berechtigungen wurden hinzugefügt zu. CloudWatchReadOnlyAccess Bestehende Nur-Lese-Berechtigungen für CloudWatch Synthetics-, X-Ray- und CloudWatch RUM-Aktionen sowie neue Nur-Lese-Berechtigungen für CloudWatch Application Signals wurden hinzugefügt, sodass Benutzer mit dieser Richtlinie ihre von Application Signals gemeldeten Dienstintegritätsprobleme analysieren und diagnostizieren können. CloudWatch Die `cloudwatch:GenerateQuery` Berechtigung wurde hinzugefügt, damit Benutzer mit dieser Richtlinie anhand einer Aufforderung in natürlicher Sprache eine CloudWatch Metrics Insights-Abfrageze ichenfolge generieren können.	05. Dezember 2023

Änderung	Beschreibung	Datum
CloudWatchApplicationSignalsServiceRolePolicy – Neue Richtlinie.	CloudWatch hat eine neue Richtlinie hinzugefügt CloudWatchApplicationSignalsServiceRolePolicy. Das CloudWatchApplicationSignalsServiceRolePolicy gewährt einer kommenden Funktion Berechtigungen zum Sammeln von CloudWatch Logdaten, X-Ray-Trace-Daten, CloudWatch Metrikdaten und Tagging-Daten.	9. November 2023
AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicy – Neue Richtlinie.	CloudWatch hat eine neue Richtlinie AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicyhinzugefügt. Das AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicyerteilt die Erlaubnis, in CloudWatch Ihrem Namen Performance Insights Insights-Metriken aus Datenbanken abzurufen.	20. September 2023

Änderung	Beschreibung	Datum
CloudWatchReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	CloudWatch hat eine Erlaubnis zu CloudWatchReadOnlyAccesshinzugefügt. Die `application-autoscaling:DescribeScalingPolicies` -Berechtigung wurde hinzugefügt, damit Benutzer mit dieser Richtlinie auf Informationen über Richtlinien für Application Auto Scaling zugreifen können.	14. September 2023
CloudWatchFullAccessV2 — Neue Richtlinie	CloudWatch hat eine neue Richtlinie CloudWatchFullAccessV2 hinzugefügt. Die CloudWatchFullAccessV2 gewährt vollen Zugriff auf CloudWatch Aktionen und Ressourcen und bietet gleichzeitig einen besseren Umfang der Berechtigungen, die anderen Diensten wie Amazon SNS und gewährt wurden. Amazon EC2 Auto ScalingWeitere Informationen finden Sie unter V2. CloudWatchFullAccess	1. August 2023

Änderung	Beschreibung	Datum
AWSServiceRoleForInternetMonitor – Aktualisierung auf eine bestehende Richtlinie	Amazon CloudWatch Internet Monitor hat neue Berechtigungen zur Überwachung von Network Load Balancer Balancer-Ressourcen hinzugefügt. Die Berechtigungen `elasticloadbalancing:DescribeLoadBalancers` und `ec2:DescribeNetworkInterfaces` sind erforderlich, damit Internet Monitor den Datenverkehr des Network Load Balancers der Kunden überwachen kann, indem es die Flussprotokolle für NLB-Ressourcen analysiert. Weitere Informationen finden Sie unter Amazon CloudWatch Internet Monitor verwenden.	15. Juli 2023

Änderung	Beschreibung	Datum
CloudWatchReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	CloudWatch hat Berechtigungen für hinzugefügt. CloudWatchReadOnlyAccess Die `logs:StopLiveTail` Berechtigungen `logs:StartLiveTail` und wurden hinzugefügt, sodass Benutzer mit dieser Richtlinie die Konsole verwenden können, um CloudWatch Logs-Live-Tail-Sitzungen zu starten und zu beenden. Weitere Informationen finden Sie unter Use live tail to view logs in near real time.	6. Juni 2023
CloudWatchCrossAccountSharingConfiguration – Neue Richtlinie.	CloudWatch hat eine neue Richtlinie hinzugefügt, mit der Sie CloudWatch kontoübergreifende Observability-Links verwalten können, die Metriken teilen CloudWatch . Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.	27. November 2022

Änderung	Beschreibung	Datum
OAM FullAccess — Neue Richtlinie	CloudWatch Es wurde eine neue Richtlinie hinzugefügt, mit der Sie CloudWatch kontenübergreifende Observability-Links und -Senken vollständig verwalten können. Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.	27. November 2022
OAM ReadOnlyAccess — Neue Richtlinie	CloudWatch Es wurde eine neue Richtlinie hinzugefügt, mit der Sie Informationen über CloudWatch kontoübergreifende Observability-Links und -Senken einsehen können. Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.	27. November 2022

Änderung	Beschreibung	Datum
CloudWatchFullAccess – Aktualisierung auf eine bestehende Richtlinie	CloudWatch hat Berechtigungen hinzugefügt zu. CloudWatchFullAccess Die `oam:ListAttachedLinks` Berechtigungen `oam:ListSinks` und wurden hinzugefügt, sodass Benutzer mit dieser Richtlinie die Konsole verwenden können, um Daten, die von Quellkonten gemeinsam genutzt wurden, CloudWatch kontenübergreifend anzusehen.	27. November 2022
CloudWatchReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	CloudWatch Berechtigungen wurden hinzugefügt zu. CloudWatchReadOnlyAccess Die `oam:ListAttachedLinks` Berechtigungen `oam:ListSinks` und wurden hinzugefügt, sodass Benutzer mit dieser Richtlinie die Konsole verwenden können, um Daten, die von Quellkonten gemeinsam genutzt wurden, CloudWatch kontenübergreifend anzusehen.	27. November 2022

Änderung	Beschreibung	Datum
AmazonCloudWatchRUMServiceRolePolicy — Aktualisierung einer bestehenden Richtlinie	CloudWatch RUM hat einen Bedingungsschlüssel in AmazonCloudWatchRUM aktualisiertServiceRolePolicy. Der `"Condition": { "StringEquals": { "cloudwatch:namespace": "AWS/RUM" } }` Bedingungsschlüssel wurde wie folgt geändert, sodass CloudWatch RUM benutzerdefinierte Metriken an benutzerdefinierte Metrik-Namespaces senden kann. `"Condition": { "StringLike": { "cloudwatch:namespace": [ "RUM/CustomMetrics/*", "AWS/RUM" ] } }`	2. Februar 2023

Änderung	Beschreibung	Datum
AmazonCloudWatchRUMReadOnlyAccess — Aktualisierte Richtlinie	CloudWatch Der AmazonCloudWatchReadOnlyAccessRUM-Richtlinie wurden Berechtigungen hinzugefügt. Die `rum:BatchGetRumMetricsDefinitions` Berechtigungen `rum:ListRumMetricsDestinations` und wurden hinzugefügt, sodass CloudWatch RUM erweiterte Messwerte an CloudWatch und Evidently senden kann.	27. Oktober 2022
AmazonCloudWatchRUMServiceRolePolicy — Aktualisierung einer bestehenden Richtlinie	CloudWatch RUM hat RUM Berechtigungen hinzugefügt AmazonCloudWatchServiceRolePolicy. Die `cloudwatch:PutMetricData` Berechtigung wurde hinzugefügt, damit CloudWatch RUM erweiterte Metriken an senden kann CloudWatch.	26. Oktober 2022

Änderung	Beschreibung	Datum
CloudWatchEvidentlyReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	CloudWatch Offensichtlich wurden Berechtigungen zu CloudWatchEvidentlyReadOnlyAccesshinzugefügt. Die Berechtigungen `evidently:GetSegment` , `evidently:ListSegments` und `evidently:ListSegmentReferences` wurden hinzugefügt, damit Benutzern mit dieser Richtlinie Evidently-Zielgruppensegmente angezeigt werden, die erstellt wurden.	12. August 2022
CloudWatchSyntheticsFullAccess – Aktualisierung auf eine bestehende Richtlinie	CloudWatch Synthetics hat Berechtigungen hinzugefügt CloudWatchSyntheticsFullAccess. Die `lambda:DeleteLayerVersion` Berechtigungen `lambda:DeleteFunction` und wurden hinzugefügt, sodass CloudWatch Synthetics verwandte Ressourcen löschen kann, wenn ein Canary gelöscht wird. `iam:ListAttachedRolePolicies` wurde hinzugefügt, damit Kunden die Richtlinien einsehen können, die an die IAM-Rolle eines Canarys angehängt sind.	6. Mai 2022

Änderung	Beschreibung	Datum
AmazonCloudWatchRUMFullAccess — Neue Richtlinie	CloudWatch Es wurde eine neue Richtlinie hinzugefügt, um die vollständige Verwaltung von CloudWatch RUM zu ermöglichen. CloudWatch Mit RUM können Sie eine echte Benutzerüberwachung Ihrer Webanwendung durchführen. Weitere Informationen finden Sie unter Verwenden Sie CloudWatch RUM.	29. November 2021
AmazonCloudWatchRUMReadOnlyAccess — Neue Richtlinie	CloudWatch Es wurde eine neue Richtlinie hinzugefügt, um den schreibgeschützten Zugriff auf RUM zu CloudWatch ermöglichen. CloudWatch Mit RUM können Sie eine echte Benutzerüberwachung Ihrer Webanwendung durchführen. Weitere Informationen finden Sie unter Verwenden Sie CloudWatch RUM.	29. November 2021

Änderung	Beschreibung	Datum
CloudWatchEvidentlyFullAccess – Neue Richtlinie.	CloudWatch hat eine neue Richtlinie hinzugefügt, um die vollständige Verwaltung von CloudWatch Evidently zu ermöglichen. CloudWatch Ermöglicht es Ihnen, A/B-Experimente mit Ihren Webanwendungen durchzuführen und diese schrittweise einzuführen. Weitere Informationen finden Sie unter Führen Sie Produkteinführungen und A/B-Experimente mit CloudWatch Eviently durch.	29. November 2021
CloudWatchEvidentlyReadOnlyAccess – Neue Richtlinie.	CloudWatch hat eine neue Richtlinie hinzugefügt, um den schreibgeschützten Zugriff auf Eviently zu ermöglichen. CloudWatch CloudWatch Ermöglicht es Ihnen, A/B-Experimente mit Ihren Webanwendungen durchzuführen und diese schrittweise einzuführen. Weitere Informationen finden Sie unter Führen Sie Produkteinführungen und A/B-Experimente mit CloudWatch Eviently durch.	29. November 2021

Änderung	Beschreibung	Datum
AWSServiceRoleForCloudWatchRUM— Neue verwaltete Richtlinie	CloudWatch Es wurde eine Richtlinie für eine neue dienstbezogene Rolle hinzugefügt, die es CloudWatch RUM ermöglicht, Überwachungsdaten für andere relevante AWS Dienste zu veröffentlichen.	29. November 2021

Änderung	Beschreibung	Datum
CloudWatchSyntheticsFullAccess – Aktualisierung auf eine bestehende Richtlinie	CloudWatch Synthetics hat Berechtigungen zu CloudWatchSyntheticsFullAccesseiner Berechtigung hinzugefügt und auch deren Umfang geändert. Die `kms:ListAliases` Berechtigung wurde hinzugefügt, damit Benutzer verfügbare AWS KMS Schlüssel auflisten können, mit denen kanarische Artefakte verschlüsselt werden können. Die Berechtigung `kms:DescribeKey` kam hinzu, damit Benutzer Details zu den Schlüsseln sehen können, die zum Verschlüsseln von Canary-Artefakten verwendet werden. Und die Berechtigung `kms:Decrypt` ermöglicht es Benutzern, Canary-Artefakte zu entschlüsseln. Diese Fähigkeit zur Entschlüsselung ist auf Ressourcen in Amazon S3 Buckets beschränkt. Der `Resource`-Scope der Berechtigung `s3:GetBucketLocation` wurde von `*` zu `arn:aws:s3:::*` geändert.	29. September 2021

Änderung	Beschreibung	Datum
CloudWatchSyntheticsFullAccess – Aktualisierung auf eine bestehende Richtlinie	CloudWatch Synthetics hat eine Genehmigung zu CloudWatchSyntheticsFullAccesshinzugefügt. Die Berechtigung `lambda:UpdateFunctionCode` wurde hinzugefügt, damit Benutzer mit dieser Richtlinie die Laufzeitversion von Canaries ändern können.	20. Juli 2021
AWSCloudWatchAlarms_ActionSSMIncidentsServiceRolePolicy— Neue verwaltete Richtlinie	CloudWatch Es wurde eine neue verwaltete IAM-Richtlinie hinzugefügt, mit der Vorfälle im AWS Systems Manager Incident Manager erstellt werden können CloudWatch .	10. Mai 2021
CloudWatchAutomaticDashboardsAccess – Aktualisierung auf eine bestehende Richtlinie	CloudWatch hat der CloudWatchAutomaticDashboardsAccessverwalteten Richtlinie eine Berechtigung hinzugefügt. Die `synthetics:DescribeCanariesLastRun` Berechtigung wurde zu dieser Richtlinie hinzugefügt, damit kontoübergreifende Dashboard-Benutzer Details zu CloudWatch Synthetics Canary Runs einsehen können.	20. April 2021

Änderung	Beschreibung	Datum
CloudWatch hat begonnen, Änderungen zu verfolgen	CloudWatch hat begonnen, Änderungen für die AWS verwalteten Richtlinien zu verfolgen.	14. April 2021

Verwendung von Bedingungsschlüsseln zur Beschränkung des Zugriffs auf CloudWatch Namespaces

Verwenden Sie IAM-Bedingungsschlüssel, um Benutzer darauf zu beschränken, Metriken nur in den von Ihnen angegebenen CloudWatch Namespaces zu veröffentlichen.

Zulassen des Veröffentlichens in nur einem Namespace

Die folgende Richtlinie schränkt den Benutzer auf das Veröffentlichen von Metriken im Namespace mit dem Namen `MyCustomNamespace` ein.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Resource": "*",
        "Action": "cloudwatch:PutMetricData",
        "Condition": {
            "StringEquals": {
                "cloudwatch:namespace": "MyCustomNamespace"
            }
        }
    }
}
```

Ausschließen des Veröffentlichens aus einem Namespace

Die folgende Richtlinie ermöglicht dem Benutzer das Veröffentlichen von Metriken in jedem Namespace mit Ausnahme von `CustomNamespace2`.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData"
        },
        {
            "Effect": "Deny",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CustomNamespace2"
                }
            }
        }
    ]
}
```

Verwenden von Bedingungsschlüsseln, um den Zugriff von Contributor-Insights-Benutzern auf Protokollgruppen einzuschränken

Um eine Regel in Contributor Insights zu erstellen und ihre Ergebnisse anzuzeigen, muss ein Benutzer die Berechtigung `cloudwatch:PutInsightRule` haben. Standardmäßig kann ein Benutzer mit dieser Berechtigung eine Contributor Insights-Regel erstellen, die jede Protokollgruppe in CloudWatch Logs auswertet und dann die Ergebnisse anzeigt. Die Ergebnisse können Contributor-Daten für diese Protokollgruppen enthalten.

Sie können IAM-Richtlinien mit Bedingungsschlüsseln erstellen, um Benutzern die Berechtigung zum Schreiben von Contributor-Insights-Regeln für einige Protokollgruppen zu erteilen und gleichzeitig zu verhindern, dass sie Regeln für andere Protokollgruppen schreiben und diese Daten aus anderen Protokollgruppen anzeigen.

Weitere Informationen zum Element `Condition` in IAM-Richtlinien finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung.

Zugriff auf Schreibregeln und Anzeigen von Ergebnissen nur für bestimmte Protokollgruppen zulassen

Die folgende Richtlinie ermöglicht dem Benutzer den Zugriff zum Schreiben von Regeln und Anzeigen von Ergebnissen für die Protokollgruppe mit dem Namen `AllowedLogGroup` und alle

Protokollgruppen, deren Namen mit `AllowedWildCard` beginnen. Sie gewährt keinen Zugriff auf Schreibregeln oder das Anzeigen von Regelergebnissen für andere Protokollgruppen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCertainLogGroups",
            "Effect": "Allow",
            "Action": "cloudwatch:PutInsightRule",
            "Resource": "arn:aws:cloudwatch:*:*:insight-rule/*",
            "Condition": {
                "ForAllValues:StringEqualsIgnoreCase": {
                    "cloudwatch:requestInsightRuleLogGroups": [
                        "AllowedLogGroup",
                        "AllowedWildcard*"
                    ]
                }
            }
        }
    ]
}
```

Schreibregeln für bestimmte Protokollgruppen verweigern, aber Schreibregeln für alle anderen Protokollgruppen zulassen

Die folgende Richtlinie verweigert dem Benutzer explizit den Zugriff auf das Schreiben von Regeln und das Anzeigen von Regelergebnissen für die Protokollgruppe mit dem Namen `ExplicitlyDeniedLogGroup`, erlaubt jedoch das Schreiben von Regeln und das Anzeigen von Regelergebnissen für alle anderen Protokollgruppen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowInsightRulesOnLogGroupsByDefault",
            "Effect": "Allow",
            "Action": "cloudwatch:PutInsightRule",
            "Resource": "arn:aws:cloudwatch:*:*:insight-rule/*"

        },
        {
```

```
            "Sid": "ExplicitDenySomeLogGroups",
            "Effect": "Deny",
            "Action": "cloudwatch:PutInsightRule",
            "Resource": "arn:aws:cloudwatch:*:*:insight-rule/*",
            "Condition": {
                "ForAllValues:StringEqualsIgnoreCase": {
                    "cloudwatch:requestInsightRuleLogGroups": [
                        "/test/alpine/ExplicitlyDeniedLogGroup"
                    ]
                }
            }
        }
    ]
}
```

Verwenden von Bedingungsschlüsseln zum Begrenzen von Alarmaktionen

Wenn CloudWatch Alarme ihren Status ändern, können sie verschiedene Aktionen ausführen, z. B. das Stoppen und Beenden von EC2-Instances und das Ausführen von Systems Manager Manager-Aktionen. Diese Aktionen können ausgelöst werden, wenn sich der Alarm in einen beliebigen Zustand ändert, einschließlich ALARM, OK oder INSUPFIZIENT_DATA.

Verwenden Sie den `cloudwatch:AlarmActions`-Bedingungsschlüssel, um einem Benutzer zu ermöglichen, Alarme zu erstellen, die nur die von Ihnen angegebenen Aktionen ausführen können, wenn sich der Alarmstatus ändert. Beispielsweise können Sie einem Benutzer erlauben, Alarme zu erstellen, die nur Aktionen ausführen können, die keine EC2-Aktionen sind.

Einem Benutzer erlauben, Alarme zu erstellen, die nur Amazon-SNS-Benachrichtigungen senden oder Systems-Manager-Aktionen ausführen können

Die folgende Richtlinie beschränkt den Benutzer darauf, Alarme zu erstellen, die nur Amazon-SNS-Benachrichtigungen senden und Systems-Manager-Aktionen ausführen können. Der Benutzer kann keine Alarme erstellen, die EC2-Aktionen ausführen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateAlarmsThatCanPerformOnlySNSandSSMActions",
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricAlarm",
```

```
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringLike": {
                    "cloudwatch:AlarmActions": [
                        "arn:aws:sns:*",
                        "arn:aws:ssm:*"
                    ]
                }
            }
        }
    ]
}
```

Verwenden von serviceverknüpften Rollen für CloudWatch

Amazon CloudWatch verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist eine einzigartige Art von IAM-Rolle, mit der direkt verknüpft ist. CloudWatch Mit Diensten verknüpfte Rollen sind vordefiniert CloudWatch und enthalten alle Berechtigungen, die der Dienst benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen.

Eine servicebezogene Rolle CloudWatch ermöglicht das Einrichten von CloudWatch Alarmen, mit denen eine Amazon EC2 EC2-Instance beendet, gestoppt oder neu gestartet werden kann, ohne dass Sie die erforderlichen Berechtigungen manuell hinzufügen müssen. Eine weitere servicebezogene Rolle ermöglicht es einem Monitoring-Konto, auf CloudWatch Daten von anderen Konten zuzugreifen, die Sie angeben, um kontoübergreifende, regionsübergreifende Dashboards zu erstellen.

CloudWatch definiert die Berechtigungen dieser dienstbezogenen Rollen und kann, sofern nicht anders definiert, nur CloudWatch die Rolle übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können die Rollen nur nach dem Löschen der zugehörigen Ressourcen löschen. Diese Einschränkung schützt Ihre CloudWatch Ressourcen, da Sie die Berechtigungen für den Zugriff auf die Ressourcen nicht versehentlich entfernen können.

Informationen zu anderen Services, die serviceverknüpften Rollen unterstützen, finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Dienstbezogene Rollenberechtigungen für Alarme, EC2-Aktionen CloudWatch

CloudWatch verwendet die serviceverknüpfte Rolle mit dem Namen AWSServiceRoleForCloudWatchEvents— CloudWatch verwendet diese serviceverknüpfte Rolle, um Amazon EC2 EC2-Alarmaktionen durchzuführen.

Die AWSServiceRoleForCloudWatchEvents serviceverknüpfte Rolle vertraut darauf, dass der CloudWatch Events-Service die Rolle übernimmt. CloudWatch Events ruft die Aktionen zum Beenden, Stoppen oder Neustarten der Instance auf, wenn sie durch den Alarm ausgelöst werden.

Die Richtlinie für AWSServiceRoleForCloudWatchEvents servicebezogene Rollenberechtigungen ermöglicht es CloudWatch Events, die folgenden Aktionen auf Amazon EC2 EC2-Instances durchzuführen:

- `ec2:StopInstances`
- `ec2:TerminateInstances`
- `ec2:RecoverInstances`
- `ec2:DescribeInstanceRecoveryAttribute`
- `ec2:DescribeInstances`
- `ec2:DescribeInstanceStatus`

Die Richtlinie für AWSServiceRoleForCloudWatchCrossAccountdienstbezogene Rollenberechtigungen ermöglicht die Durchführung CloudWatch der folgenden Aktionen:

- `sts:AssumeRole`

Dienstbezogene Rollenberechtigungen für Anwendungssignale CloudWatch

CloudWatch Application Signals verwendet die dienstverknüpfte Rolle mit dem Namen AWSServiceRoleForCloudWatchApplicationSignals— CloudWatch verwendet diese dienstverknüpfte Rolle, um CloudWatch Logdaten, X-Ray-Trace-Daten, CloudWatch Metrikdaten und Tagging-Daten von Anwendungen zu sammeln, die Sie für CloudWatch Application Signals aktiviert haben.

Die AWSServiceRoleForCloudWatchApplicationSignalsdienstbezogene Rolle vertraut darauf, dass CloudWatch Application Signals die Rolle übernimmt. Application Signals erfasst die Protokoll-, Trace-, Metrik- und Tag-Daten aus Ihrem Konto.

Der AWSServiceRoleForCloudWatchApplicationSignalsist eine IAM-Richtlinie angehängt, und diese Richtlinie trägt den Namen. CloudWatchApplicationSignalsServiceRolePolicy Diese Richtlinie erteilt CloudWatch Application Signals die Erlaubnis, Überwachungs- und Kennzeichnungsdaten von anderen relevanten AWS Diensten zu sammeln. Sie enthält Berechtigungen, die Application Signals die folgenden Aktionen ermöglichen:

- `xray:GetServiceGraph`
- `logs:StartQuery`
- `logs:GetQueryResults`
- `cloudwatch:GetMetricData`
- `cloudwatch:ListMetrics`
- `tag:GetResources`

Der vollständige Inhalt von CloudWatchApplicationSignalsServiceRolePolicylautet wie folgt:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "XRayPermission",
            "Effect": "Allow",
            "Action": [
                "xray:GetServiceGraph"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "CWLogsPermission",
            "Effect": "Allow",
            "Action": [
                "logs:StartQuery",
                "logs:GetQueryResults"
            ],
```

```
            "Resource": [
                "arn:aws:logs:*:*:log-group:/aws/appsignals/*:*",
                "arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "CWListMetricsPermission",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:ListMetrics"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "CWGetMetricDataPermission",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "TagsPermission",
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": [
                "*"
            ],
```

```
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        }
    ]
}
```

Dienstbezogene Rollenberechtigungen für CloudWatch Alarme Systems Manager Manager-Aktionen OpsCenter

CloudWatch verwendet die angegebene dienstverknüpfte Rolle AWSServiceRoleForCloudWatchAlarms_ActionSSM— CloudWatch verwendet diese dienstverknüpfte Rolle, um Systems Manager OpsCenter Manager-Aktionen auszuführen, wenn ein CloudWatch Alarm in den ALARM-Status wechselt.

Die AWSServiceRoleForCloudWatchAlarms_ActionSSM dienstverknüpfte Rolle vertraut darauf, dass der CloudWatch Dienst die Rolle übernimmt. CloudWatch Alarme rufen die Systems Manager OpsCenter Manager-Aktionen auf, wenn sie vom Alarm ausgelöst werden.

Die Richtlinie für AWSServiceRoleForCloudWatchAlarms_ActionSSMdienstbezogene Rollenberechtigungen ermöglicht es Systems Manager, die folgenden Aktionen durchzuführen:

- `ssm:CreateOpsItem`

Dienstbezogene Rollenberechtigungen für CloudWatch Alarme Systems Manager Incident Manager-Aktionen

CloudWatch verwendet die mit dem Dienst verknüpfte Rolle mit dem Namen AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidents— CloudWatch verwendet diese dienstbezogene Rolle, um Incident Manager-Vorfälle auszulösen, wenn ein CloudWatch Alarm in den ALARM-Status wechselt.

Die AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidentsserviceverknüpfte Rolle vertraut darauf, dass der CloudWatch Dienst die Rolle übernimmt. CloudWatch Alarme rufen die Aktion Systems Manager Incident Manager auf, wenn sie vom Alarm ausgelöst werden.

Die Richtlinie für AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidentsdienstbezogene Rollenberechtigungen ermöglicht es Systems Manager, die folgenden Aktionen durchzuführen:

- `ssm-incidents:StartIncident`

Dienstbezogene Rollenberechtigungen für CloudWatch kontoübergreifende, regionsübergreifende

CloudWatch verwendet die angegebene dienstbezogene Rolle AWSServiceRoleForCloudWatchCrossAccount— CloudWatch verwendet diese Rolle, um auf CloudWatch Daten in anderen von Ihnen angegebenen AWS Konten zuzugreifen. Die SLR gewährt lediglich die Berechtigung „Rolle übernehmen“, damit der CloudWatch Dienst die Rolle im Sharing-Konto übernehmen kann. Es ist die Freigaberolle, die den Zugriff auf Daten ermöglicht.

Die mit dem AWSServiceRoleForCloudWatchCrossAccountDienst verknüpfte Richtlinie für Rollenberechtigungen ermöglicht CloudWatch die Durchführung der folgenden Aktionen:

- `sts:AssumeRole`

Die AWSServiceRoleForCloudWatchCrossAccountdienstverknüpfte Rolle vertraut darauf, dass der CloudWatch Dienst die Rolle übernimmt.

Dienstbezogene Rollenberechtigungen für CloudWatch Datenbank Performance Insights

CloudWatch verwendet die benannte dienstverknüpfte Rolle. AWSServiceRoleForCloudWatchMetrics_DbPerfInsights — CloudWatch verwendet diese Rolle, um Performance Insights Insights-Metriken für die Erstellung von Alarmen und Snapshots abzurufen.

Der AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsserviceverknüpften Rolle ist die `AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicy` IAM-Richtlinie angehängt. Der Inhalt dieser Richtlinie lautet wie folgt:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "pi:GetResourceMetrics"
   ],
   "Resource": "*",
```

```
    "Condition": {
     "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
     }
    }
   }
  ]
}
```

Die AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsdienstbezogene Rolle vertraut darauf, dass der CloudWatch Dienst die Rolle übernimmt.

Erstellen einer dienstbezogenen Rolle für CloudWatch

Sie müssen keine dieser serviceverknüpften Rollen manuell erstellen. Wenn Sie zum ersten Mal einen Alarm in der erstellen AWS Management Console, CloudWatch erstellt die IAM-CLI oder die IAM-API AWSServiceRoleForCloudWatchEvents und AWSServiceRoleForCloudWatchAlarms_ActionSSMfür Sie.

Wenn Sie die Service- und Topologieerkennung zum ersten Mal aktivieren, erstellt AWSServiceRoleForCloudWatchApplicationSignalsApplication Signals diese für Sie.

Wenn Sie ein Konto zum ersten Mal als Überwachungskonto für kontenübergreifende regionsübergreifende Funktionen aktivieren, CloudWatch erstellt AWSServiceRoleForCloudWatchCrossAccountes für Sie.

Wenn Sie zum ersten Mal einen Alarm erstellen, der die `DB_PERF_INSIGHTS` metrische mathematische Funktion verwendet, CloudWatch wird er AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsfür Sie erstellt.

Weitere Informationen finden Sie unter Erstellen einer serviceverknüpfte Rolle im IAM-Leitfaden.

Bearbeiten einer serviceverknüpften Rolle für CloudWatch

CloudWatch erlaubt es Ihnen nicht, die AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsRollen AWSServiceRoleForCloudWatchEvents, AWSServiceRoleForCloudWatchAlarms_ActionSSMAWSServiceRoleForCloudWatchCrossAccount, oder zu bearbeiten. Nachdem Sie diese Rollen erstellt haben, können Sie ihre Namen nicht ändern, da verschiedene Entitäten möglicherweise auf diese Rollen verweisen. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten.

Bearbeiten der Beschreibung einer serviceverknüpften Rolle (IAM-Konsole)

Sie können die IAM-Konsole für das Bearbeiten der Beschreibung einer serviceverknüpften Rolle verwenden.

So bearbeiten Sie die Beschreibung einer serviceverknüpften Rolle (Konsole)

1. Wählen Sie im Navigationsbereich der IAM Console Roles (Rollen) aus.
2. Wählen Sie den Namen der zu ändernden Rolle.
3. Wählen Sie neben Role description ganz rechts Edit.
4. Geben Sie die neue Beschreibung im Dialogfeld ein und klicken Sie auf Save (Speichern).

Bearbeiten der Beschreibung einer serviceverknüpften Rolle (AWS CLI)

Sie können die IAM-Befehle von verwenden AWS Command Line Interface , um die Beschreibung einer serviceverknüpften Rolle zu bearbeiten.

So ändern Sie die Beschreibung einer serviceverknüpften Rolle (AWS CLI)

1. (Optional) Um die aktuelle Beschreibung einer Rolle anzuzeigen, verwenden Sie die folgenden Befehle:

   ```
   $ aws iam get-role --role-name role-name
   ```

   Verwenden Sie den Rollennamen, nicht den ARN, um sich auf Rollen mit den AWS CLI -Befehlen zu beziehen. Wenn eine Rolle zum Beispiel folgenden ARN hat: `arn:aws:iam::123456789012:role/myrole`, verweisen Sie auf die Rolle als **`myrole`**.
2. Um die Beschreibung einer serviceverknüpften Rolle zu aktualisieren, verwenden Sie den folgenden Befehl:

   ```
   $ aws iam update-role-description --role-name role-name --description description
   ```

Bearbeiten der Beschreibung einer serviceverknüpften Rolle (IAM-API)

Sie können die IAM-API für das Bearbeiten der Beschreibung einer serviceverknüpften Rolle verwenden.

So ändern Sie die Beschreibung einer serviceverknüpften Rolle (API)

1. (Optional) Um die aktuelle Beschreibung einer Rolle anzuzeigen, verwenden Sie den folgenden Befehl:

   GetRole
2. Um die Beschreibung einer Rolle zu aktualisieren, verwenden Sie den folgenden Befehl:

   UpdateRoleDescription

Löschen einer serviceverknüpften Rolle für CloudWatch

Wenn Sie keine Alarme mehr haben, die EC2-Instances automatisch beenden, beenden oder neu starten, empfehlen wir Ihnen, die Rolle zu löschen. AWSServiceRoleForCloudWatchEvents

Wenn Sie keine Alarme mehr haben, die Systems Manager OpsCenter Manager-Aktionen ausführen, empfehlen wir Ihnen, die AWSServiceRoleForCloudWatchAlarms_ActionSSM Rolle zu löschen.

Wenn Sie alle Alarme löschen, die die `DB_PERF_INSIGHTS` metrische mathematische Funktion verwenden, empfehlen wir Ihnen, die mit dem AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsDienst verknüpfte Rolle zu löschen.

Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie löschen können.

Bereinigen einer serviceverknüpften Rolle

Bevor Sie mit IAM eine serviceverknüpfte Rolle löschen können, müssen Sie sich zunächst vergewissern, dass die Rolle über keine aktiven Sitzungen verfügt, und alle Ressourcen entfernen, die von der Rolle verwendet werden.

So überprüfen Sie in der IAM-Konsole, ob die serviceverknüpfte Rolle über eine aktive Sitzung verfügt

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Rollen aus. Wählen Sie den Namen (nicht das Kontrollkästchen) der AWSServiceRoleForCloudWatchEvents Rolle.
3. Wählen Sie auf der Seite Summary der ausgewählten Rolle Access Advisor und überprüfen Sie die letzten Aktivitäten auf die serviceverknüpfte Rolle.

> **Note**
>
> Wenn Sie sich nicht sicher sind, ob die AWSServiceRoleForCloudWatchEvents Rolle verwendet CloudWatch wird, versuchen Sie, die Rolle zu löschen. Wenn der Service die Rolle verwendet, schlägt die Löschung fehl und Sie können die -Regionen anzeigen, in denen die Rolle verwendet wird. Wenn die Rolle verwendet wird, müssen Sie warten, bis die Sitzung beendet wird, bevor Sie die Rolle löschen können. Die Sitzung für eine serviceverknüpfte Rolle können Sie nicht widerrufen.

Löschen einer serviceverknüpften Rolle (IAM-Konsole)

Sie können die IAM-Konsole für das Löschen einer serviceverknüpften Rolle verwenden.

So löschen Sie eine serviceverknüpfte Rolle (Konsole)

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Rollen aus. Aktivieren Sie dann das Kontrollkästchen neben dem Namen der Rolle, die Sie löschen möchten, nicht den Namen oder die Zeile selbst.
3. Klicken Sie bei Role actions auf Delete role.
4. Überprüfen Sie im Bestätigungsdialogfeld die letzten Service-Zugriffsdaten, die zeigen, wann jede der ausgewählten Rollen zuletzt auf den AWS -Service zugegriffen hat. Auf diese Weise können Sie leichter bestätigen, ob die Rolle derzeit aktiv ist. Wählen Sie Yes, Delete, um fortzufahren.
5. Sehen Sie sich die Benachrichtigungen in der IAM-Konsole an, um den Fortschritt der Löschung der serviceverknüpften Rolle zu überwachen. Da die Löschung der serviceverknüpften IAM-Rolle asynchron erfolgt, kann die Löschung nach dem Übermitteln der Rolle für die Löschung erfolgreich sein oder fehlschlagen. Wenn der Vorgang fehlschlägt, wählen Sie in den Benachrichtigungen View details oder View Resources aus, um zu erfahren, warum die Löschung fehlgeschlagen ist. Wenn das Löschen fehlschlägt, weil der Service Ressourcen enthält, die von der Rolle verwendet werden, enthält die Angabe des Fehlergrundes eine Liste der Ressourcen.

Löschen einer serviceverknüpften Rolle (AWS CLI)

Sie können IAM-Befehle von verwenden, AWS Command Line Interface um eine dienstverknüpfte Rolle zu löschen.

So löschen Sie eine serviceverknüpfte Rolle (AWS CLI)

1. Da eine serviceverknüpfte Rolle nicht gelöscht werden kann, wenn sie verwendet wird oder ihr Ressourcen zugeordnet sind, müssen Sie eine Löschungsanforderung übermitteln. Diese Anforderung kann verweigert werden, wenn diese Bedingungen nicht erfüllt sind. Sie benötigen die `deletion-task-id` aus der Antwort, um den Status der Löschaufgabe zu überprüfen. Geben Sie den folgenden Befehl ein, um eine Löschanforderung für eine serviceverknüpfte Rolle zu übermitteln:

   ```
   $ aws iam delete-service-linked-role --role-name service-linked-role-name
   ```

2. Geben Sie den folgenden Befehl ein, um den Status der Löschaufgabe zu überprüfen:

   ```
   $ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
   ```

   Der Status der Löschaufgabe kann `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` oder `FAILED` lauten. Wenn die Löschung fehlschlägt, gibt der Aufruf den Grund zurück, sodass Sie das Problem beheben können.

Löschen einer serviceverknüpften Rolle (IAM-API)

Sie können die IAM-API zum Löschen einer serviceverknüpften Rolle verwenden.

So löschen Sie eine serviceverknüpfte Rolle (API)

1. Rufen Sie an, um eine Löschanfrage für eine dienstverknüpfte Rolle einzureichen. DeleteServiceLinkedRole Geben Sie in der Anforderung den Namen der Rolle an, die Sie löschen möchten.

   Da eine serviceverknüpfte Rolle nicht gelöscht werden kann, wenn sie verwendet wird oder ihr Ressourcen zugeordnet sind, müssen Sie eine Löschungsanforderung übermitteln. Diese Anforderung kann verweigert werden, wenn diese Bedingungen nicht erfüllt sind. Sie benötigen die `DeletionTaskId` aus der Antwort, um den Status der Löschaufgabe zu überprüfen.

2. Rufen GetServiceLinkedRoleDeletionStatusSie an, um den Status des Löschvorgangs zu überprüfen. Geben Sie in der Anforderung die `DeletionTaskId` an.

   Der Status der Löschaufgabe kann `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` oder `FAILED` lauten. Wenn die Löschung fehlschlägt, gibt der Aufruf den Grund zurück, sodass Sie das Problem beheben können.

CloudWatch Aktualisierungen von Rollen, die AWS mit Diensten verknüpft sind

Hier finden Sie Informationen zu Aktualisierungen der AWS verwalteten Richtlinien, die CloudWatch seit Beginn der Nachverfolgung dieser Änderungen durch diesen Dienst vorgenommen wurden. Abonnieren Sie den RSS-Feed auf der Seite CloudWatch Dokumentenverlauf, um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten.

Änderung	Beschreibung	Datum
AWSServiceRoleForCloudWatchApplicationSignals — Aktualisierung der Richtlinien für die Berechtigungen der dienstbezogenen Rollen	CloudWatch fügt dem Geltungsbereich der `logs:StartQuery` und den durch diese Rolle gewährten `logs:GetQueryResults` Berechtigungen weitere Protokollgruppen hinzu.	24. April 2024
AWSServiceRoleForCloudWatchApplicationSignals — Neue dienstbezogene Rolle	CloudWatch hat diese neue dienstbezogene Rolle hinzugefügt, damit CloudWatch Application Signals CloudWatch Logdaten, X-Ray-Trace-Daten, CloudWatch Metrikdaten und Tagging-Daten von Anwendungen sammeln kann, die Sie für	9. November 2023

Änderung	Beschreibung	Datum
	CloudWatch Application Signals aktiviert haben.	
AWSServiceRoleForCloudWatchMetrics_DbPerfInsights— Neue dienstbezogene Rolle	CloudWatch hat diese neue dienstbezogene Rolle hinzugefügt, um das Abrufen von Performance Insights Insights-Metriken für Alarme und Snapshots CloudWatch zu ermöglichen. Dieser Rolle ist eine IAM-Richtlinie zugeordnet, und die Richtlinie erteilt die Erlaubnis, Performance Insights Insights-Metriken in Ihrem Namen abzurufen. CloudWatch	13. September 2023
AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidents— Neue dienstbezogene Rolle	CloudWatch Es wurde eine neue dienstbezogene Rolle hinzugefügt, mit der Vorfälle im AWS Systems Manager Incident Manager erstellt werden können CloudWatch .	26. April 2021
CloudWatch hat begonnen, Änderungen zu verfolgen	CloudWatch hat mit der Nachverfolgung von Änderungen für seine dienstbezogenen Rollen begonnen.	26. April 2021

Verwendung von serviceverknüpften Rollen für RUM CloudWatch

CloudWatch RUM verwendet eine AWS Identity and Access Management serviceverknüpfte (IAM) -Rolle. Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit RUM verknüpft ist. Die dienstgebundene Rolle ist von RUM vordefiniert und umfasst alle Berechtigungen, die der Dienst benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen.

RUM definiert die Berechtigungen dieser serviceverknüpften Rollen. Sofern keine andere Konfiguration festgelegt wurde, kann nur RUM die Rolle übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können die Rolle nur nach dem Löschen der zugehörigen Ressourcen löschen. Dies schützt Ihre RUM-Ressourcen, da Sie nicht versehentlich die Berechtigungen für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceorientierte Rollen unterstützen, finden Sie unter AWS services that work with IAM (-Services, die mit IAM funktionieren). Suchen Sie nach den Services, für die Yes (Ja) in der Spalte Service-linked roles (Serviceorientierte Rollen) angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Berechtigungen von serviceverknüpften Rollen für RUM

RUM verwendet die mit dem Dienst verknüpfte Rolle mit dem Namen AWSServiceRoleForCloudWatchRUM— diese Rolle ermöglicht es RUM, AWS X-Ray Trace-Daten an Ihr Konto zu senden, für App-Monitore, für die Sie X-Ray Tracing aktivieren.

Die AWSServiceRoleForCloudWatchRUMdienstbezogene Rolle vertraut darauf, dass der X-Ray-Dienst die Rolle übernimmt. X-Ray sendet die Nachverfolgungsdaten an Ihr Konto.

Der AWSServiceRoleForCloudWatchRUMserviceverknüpften Rolle ist eine IAM-Richtlinie mit dem Namen RUM angehängt. AmazonCloudWatch ServiceRolePolicy Diese Richtlinie erteilt CloudWatch RUM die Erlaubnis, Überwachungsdaten für andere relevante AWS Dienste zu veröffentlichen. Sie enthält Berechtigungen, die RUM die folgenden Aktionen ermöglichen:

- `xray:PutTraceSegments`
- `cloudwatch:PutMetricData`

Der vollständige Inhalt von AmazonCloudWatchRUM ServiceRolePolicy lautet wie folgt.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "xray:PutTraceSegments"
   ],
   "Resource": [
    "*"
   ]
  },
  {
   "Effect": "Allow",
   "Action": "cloudwatch:PutMetricData",
   "Resource": "*",
   "Condition": {
    "StringLike": {
     "cloudwatch:namespace": [
      "RUM/CustomMetrics/*",
      "AWS/RUM"
     ]
    }
   }
  }
 ]
}
```

Erstellen einer serviceverknüpften Rolle für RUM

Sie müssen die serviceverknüpfte Rolle für CloudWatch RUM nicht manuell erstellen. Wenn Sie zum ersten Mal einen App-Monitor mit aktiviertem X-Ray Tracing erstellen oder einen App-Monitor für die Verwendung von X-Ray Tracing aktualisieren, erstellt RUM das AWSServiceRoleForCloudWatchRUMfür Sie.

Weitere Informationen finden Sie unter Erstellen einer serviceverknüpfte Rolle im IAM-Leitfaden.

Bearbeiten einer serviceverknüpften Rolle für RUM

CloudWatch In RUM können Sie die Rolle nicht bearbeiten. AWSServiceRoleForCloudWatchRUM Nachdem Sie diese Rollen erstellt haben, können Sie ihre Namen nicht ändern, da verschiedene

Entitäten möglicherweise auf diese Rollen verweisen. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten.

Bearbeiten der Beschreibung einer serviceverknüpften Rolle (IAM-Konsole)

Sie können die IAM-Konsole für das Bearbeiten der Beschreibung einer serviceverknüpften Rolle verwenden.

So bearbeiten Sie die Beschreibung einer serviceverknüpften Rolle (Konsole)

1. Wählen Sie im Navigationsbereich der IAM Console Roles (Rollen) aus.
2. Wählen Sie den Namen der zu ändernden Rolle.
3. Wählen Sie neben Role description ganz rechts Edit.
4. Geben Sie die neue Beschreibung im Dialogfeld ein und klicken Sie auf Save (Speichern).

Bearbeiten der Beschreibung einer serviceverknüpften Rolle (AWS CLI)

Sie können die IAM-Befehle von verwenden AWS Command Line Interface , um die Beschreibung einer serviceverknüpften Rolle zu bearbeiten.

So ändern Sie die Beschreibung einer serviceverknüpften Rolle (AWS CLI)

1. (Optional) Um die aktuelle Beschreibung einer Rolle anzuzeigen, verwenden Sie die folgenden Befehle:

   ```
   $ aws iam get-role --role-name role-name
   ```

   Verwenden Sie den Rollennamen, nicht den ARN, um sich auf Rollen mit den AWS CLI -Befehlen zu beziehen. Wenn eine Rolle zum Beispiel folgenden ARN hat: `arn:aws:iam::123456789012:role/myrole`, verweisen Sie auf die Rolle als **`myrole`**.
2. Um die Beschreibung einer serviceverknüpften Rolle zu aktualisieren, verwenden Sie den folgenden Befehl:

   ```
   $ aws iam update-role-description --role-name role-name --description description
   ```

Bearbeiten der Beschreibung einer serviceverknüpften Rolle (IAM-API)

Sie können die IAM-API für das Bearbeiten der Beschreibung einer serviceverknüpften Rolle verwenden.

So ändern Sie die Beschreibung einer serviceverknüpften Rolle (API)

1. (Optional) Um die aktuelle Beschreibung einer Rolle anzuzeigen, verwenden Sie den folgenden Befehl:

   GetRole
2. Um die Beschreibung einer Rolle zu aktualisieren, verwenden Sie den folgenden Befehl:

   UpdateRoleDescription

Löschen einer serviceverknüpften Rolle für RUM

Wenn Sie keine App-Monitore mit aktiviertem X-Ray mehr haben, empfehlen wir Ihnen, die AWSServiceRoleForCloudWatchRUMRolle zu löschen.

Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie löschen können.

Bereinigen einer serviceverknüpften Rolle

Bevor Sie mit IAM eine serviceverknüpfte Rolle löschen können, müssen Sie sich zunächst vergewissern, dass die Rolle über keine aktiven Sitzungen verfügt, und alle Ressourcen entfernen, die von der Rolle verwendet werden.

So überprüfen Sie in der IAM-Konsole, ob die serviceverknüpfte Rolle über eine aktive Sitzung verfügt

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Rollen aus. Wählen Sie den Namen (nicht das Kontrollkästchen) der AWSServiceRoleForCloudWatchRUMRolle.
3. Wählen Sie auf der Seite Summary der ausgewählten Rolle Access Advisor und überprüfen Sie die letzten Aktivitäten auf die serviceverknüpfte Rolle.

> **Note**
>
> Wenn Sie sich nicht sicher sind, ob RUM die AWSServiceRoleForCloudWatchRUMRolle verwendet, versuchen Sie, die Rolle zu löschen. Wenn der Service die Rolle verwendet, schlägt die Löschung fehl und Sie können die -Regionen anzeigen, in denen die Rolle verwendet wird. Wenn die Rolle verwendet wird, müssen Sie warten, bis die Sitzung beendet wird, bevor Sie die Rolle löschen können. Die Sitzung für eine serviceverknüpfte Rolle können Sie nicht widerrufen.

Löschen einer serviceverknüpften Rolle (IAM-Konsole)

Sie können die IAM-Konsole für das Löschen einer serviceverknüpften Rolle verwenden.

So löschen Sie eine serviceverknüpfte Rolle (Konsole)

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Rollen aus. Aktivieren Sie dann das Kontrollkästchen neben dem Namen der Rolle, die Sie löschen möchten, nicht den Namen oder die Zeile selbst.
3. Klicken Sie bei Role actions auf Delete role.
4. Überprüfen Sie im Bestätigungsdialogfeld die letzten Service-Zugriffsdaten, die zeigen, wann jede der ausgewählten Rollen zuletzt auf den AWS -Service zugegriffen hat. Auf diese Weise können Sie leichter bestätigen, ob die Rolle derzeit aktiv ist. Wählen Sie Yes, Delete, um fortzufahren.
5. Sehen Sie sich die Benachrichtigungen in der IAM-Konsole an, um den Fortschritt der Löschung der serviceverknüpften Rolle zu überwachen. Da die Löschung der serviceverknüpften IAM-Rolle asynchron erfolgt, kann die Löschung nach dem Übermitteln der Rolle für die Löschung erfolgreich sein oder fehlschlagen. Wenn der Vorgang fehlschlägt, wählen Sie in den Benachrichtigungen View details oder View Resources aus, um zu erfahren, warum die Löschung fehlgeschlagen ist. Wenn das Löschen fehlschlägt, weil der Service Ressourcen enthält, die von der Rolle verwendet werden, enthält die Angabe des Fehlergrundes eine Liste der Ressourcen.

Löschen einer serviceverknüpften Rolle (AWS CLI)

Sie können IAM-Befehle von verwenden, AWS Command Line Interface um eine serviceverknüpfte Rolle zu löschen.

So löschen Sie eine serviceverknüpfte Rolle (AWS CLI)

1. Da eine serviceverknüpfte Rolle nicht gelöscht werden kann, wenn sie verwendet wird oder ihr Ressourcen zugeordnet sind, müssen Sie eine Löschungsanforderung übermitteln. Diese Anforderung kann verweigert werden, wenn diese Bedingungen nicht erfüllt sind. Sie benötigen die `deletion-task-id` aus der Antwort, um den Status der Löschaufgabe zu überprüfen. Geben Sie den folgenden Befehl ein, um eine Löschanforderung für eine serviceverknüpfte Rolle zu übermitteln:

   ```
   $ aws iam delete-service-linked-role --role-name service-linked-role-name
   ```

2. Geben Sie den folgenden Befehl ein, um den Status der Löschaufgabe zu überprüfen:

   ```
   $ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
   ```

   Der Status der Löschaufgabe kann `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` oder `FAILED` lauten. Wenn die Löschung fehlschlägt, gibt der Aufruf den Grund zurück, sodass Sie das Problem beheben können.

Löschen einer serviceverknüpften Rolle (IAM-API)

Sie können die IAM-API zum Löschen einer serviceverknüpften Rolle verwenden.

So löschen Sie eine serviceverknüpfte Rolle (API)

1. Rufen Sie an, um eine Löschanfrage für eine dienstverknüpfte Rolle einzureichen. DeleteServiceLinkedRole Geben Sie in der Anforderung den Namen der Rolle an, die Sie löschen möchten.

   Da eine serviceverknüpfte Rolle nicht gelöscht werden kann, wenn sie verwendet wird oder ihr Ressourcen zugeordnet sind, müssen Sie eine Löschungsanforderung übermitteln. Diese Anforderung kann verweigert werden, wenn diese Bedingungen nicht erfüllt sind. Sie benötigen die `DeletionTaskId` aus der Antwort, um den Status der Löschaufgabe zu überprüfen.

2. Rufen GetServiceLinkedRoleDeletionStatusSie an, um den Status des Löschvorgangs zu überprüfen. Geben Sie in der Anforderung die `DeletionTaskId` an.

   Der Status der Löschaufgabe kann `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` oder `FAILED` lauten. Wenn die Löschung fehlschlägt, gibt der Aufruf den Grund zurück, sodass Sie das Problem beheben können.

Verwenden von serviceverknüpften Rollen für CloudWatch Application Insights

CloudWatch Application Insights verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit Application Insights verknüpft ist. CloudWatch Servicebezogene Rollen sind von CloudWatch Application Insights vordefiniert und beinhalten alle Berechtigungen, die der Dienst benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen.

Eine dienstbezogene Rolle erleichtert die Einrichtung von CloudWatch Application Insights, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. CloudWatch Application Insights definiert die Berechtigungen seiner dienstbezogenen Rollen, und sofern nicht anders definiert, kann nur CloudWatch Application Insights die Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Informationen zu anderen Services, die servicegebundene Rollen unterstützen, finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte Servicegebundene Rolle angegeben ist. Wählen Sie den Link Ja, um die Dokumentation zu serviceverknüpften Rollen für diesen Service anzuzeigen.

Dienstbezogene Rollenberechtigungen für Application Insights CloudWatch

CloudWatch Application Insights verwendet die angegebene dienstverknüpfte Rolle. AWSServiceRoleForApplicationInsights Application Insights verwendet diese Rolle, um Operationen wie die Analyse der Ressourcengruppen des Kunden, die Erstellung von CloudFormation Stacks zur Erstellung von Alarmen für Metriken und die Konfiguration des CloudWatch Agenten auf EC2-Instances durchzuführen. Dieser serviceverknüpfte Rolle ist eine IAM-Richtlinie mit dem Namen `CloudwatchApplicationInsightsServiceLinkedRolePolicy` angefügt. Aktualisierungen dieser Richtlinie finden Sie unter Aktualisierungen von Application Insights auf AWS verwaltete Richtlinien.

Die Richtlinie für Rollenberechtigungen ermöglicht es CloudWatch Application Insights, die folgenden Aktionen an Ressourcen durchzuführen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:DescribeAlarmHistory",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:GetMetricData",
        "cloudwatch:ListMetrics",
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:DeleteAlarms",
        "cloudwatch:PutAnomalyDetector",
        "cloudwatch:DeleteAnomalyDetector",
        "cloudwatch:DescribeAnomalyDetectors"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:FilterLogEvents",
        "logs:GetLogEvents",
        "logs:DescribeLogStreams",
        "logs:DescribeLogGroups"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "events:DescribeRule"
      ],
      "Resource": [
        "*"
      ]
```

```
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudFormation:CreateStack",
        "cloudFormation:UpdateStack",
        "cloudFormation:DeleteStack",
        "cloudFormation:DescribeStackResources"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:*:stack/ApplicationInsights-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudFormation:DescribeStacks",
        "cloudFormation:ListStackResources",
        "cloudFormation:ListStacks"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "tag:GetResources"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "resource-groups:ListGroupResources",
        "resource-groups:GetGroupQuery",
        "resource-groups:GetGroup"
      ],
      "Resource": [
        "*"
      ]
    },
```

```
    {
      "Effect": "Allow",
      "Action": [
        "resource-groups:CreateGroup",
        "resource-groups:DeleteGroup"
      ],
      "Resource": [
        "arn:aws:resource-groups:*:*:group/ApplicationInsights-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetHealth"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:DescribeAutoScalingGroups"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:PutParameter",
        "ssm:DeleteParameter",
        "ssm:AddTagsToResource",
        "ssm:RemoveTagsFromResource",
        "ssm:GetParameters"
      ],
      "Resource": "arn:aws:ssm:*:*:parameter/AmazonCloudWatch-ApplicationInsights-*"
    },
    {
      "Effect": "Allow",
      "Action": [
```

```
        "ssm:CreateAssociation",
        "ssm:UpdateAssociation",
        "ssm:DeleteAssociation",
        "ssm:DescribeAssociation"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ssm:*:*:association/*",
        "arn:aws:ssm:*:*:managed-instance/*",
        "arn:aws:ssm:*:*:document/AWSEC2-
ApplicationInsightsCloudwatchAgentInstallAndConfigure",
        "arn:aws:ssm:*:*:document/AWS-ConfigureAWSPackage",
        "arn:aws:ssm:*:*:document/AmazonCloudWatch-ManageAgent"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetOpsItem",
        "ssm:CreateOpsItem",
        "ssm:DescribeOpsItems",
        "ssm:UpdateOpsItem",
        "ssm:DescribeInstanceInformation"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:AddTagsToResource"
      ],
      "Resource": "arn:aws:ssm:*:*:opsitem/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:ListCommandInvocations",
        "ssm:GetCommandInvocation"
      ],
      "Resource": [
        "*"
      ]
```

```
    },
    {
      "Effect": "Allow",
      "Action": "ssm:SendCommand",
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ssm:*:*:document/AWSEC2-CheckPerformanceCounterSets",
        "arn:aws:ssm:*:*:document/AWS-ConfigureAWSPackage",
        "arn:aws:ssm:*:*:document/AWSEC2-DetectWorkload",
        "arn:aws:ssm:*:*:document/AmazonCloudWatch-ManageAgent"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeVolumes",
        "ec2:DescribeVolumeStatus",
        "ec2:DescribeVpcs",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeNatGateways"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "rds:DescribeDBInstances",
        "rds:DescribeDBClusters"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "lambda:ListFunctions",
        "lambda:GetFunctionConfiguration",
        "lambda:ListEventSourceMappings"
      ],
      "Resource": [
```

```
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "events:PutRule",
        "events:PutTargets",
        "events:RemoveTargets",
        "events:DeleteRule"
      ],
      "Resource": [
        "arn:aws:events:*:*:rule/AmazonCloudWatch-ApplicationInsights-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "xray:GetServiceGraph",
        "xray:GetTraceSummaries",
        "xray:GetTimeSeriesServiceStatistics",
        "xray:GetTraceGraph"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:ListTables",
        "dynamodb:DescribeTable",
        "dynamodb:DescribeContributorInsights",
        "dynamodb:DescribeTimeToLive"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "application-autoscaling:DescribeScalableTargets"
      ],
```

```
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:GetMetricsConfiguration",
        "s3:GetReplicationConfiguration"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "states:ListStateMachines",
        "states:DescribeExecution",
        "states:DescribeStateMachine",
        "states:GetExecutionHistory"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "apigateway:GET"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DescribeClusters",
        "ecs:DescribeContainerInstances",
        "ecs:DescribeServices",
        "ecs:DescribeTaskDefinition",
        "ecs:DescribeTasks",
```

```
        "ecs:DescribeTaskSets",
        "ecs:ListClusters",
        "ecs:ListContainerInstances",
        "ecs:ListServices",
        "ecs:ListTasks"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecs:UpdateClusterSettings"
      ],
      "Resource": [
        "arn:aws:ecs:*:*:cluster/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "eks:DescribeCluster",
        "eks:DescribeFargateProfile",
        "eks:DescribeNodegroup",
        "eks:ListClusters",
        "eks:ListFargateProfiles",
        "eks:ListNodegroups",
        "fsx:DescribeFileSystems",
        "fsx:DescribeVolumes"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "sns:GetSubscriptionAttributes",
        "sns:GetTopicAttributes",
        "sns:GetSMSAttributes",
        "sns:ListSubscriptionsByTopic",
        "sns:ListTopics"
      ],
```

```
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "sqs:ListQueues"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:DeleteSubscriptionFilter"
      ],
      "Resource": [
        "arn:aws:logs:*:*:log-group:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:PutSubscriptionFilter"
      ],
      "Resource": [
        "arn:aws:logs:*:*:log-group:*",
        "arn:aws:logs:*:*:destination:AmazonCloudWatch-ApplicationInsights-
LogIngestionDestination*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:DescribeFileSystems"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "route53:GetHostedZone",
```

```
        "route53:GetHealthCheck",
        "route53:ListHostedZones",
        "route53:ListHealthChecks",
        "route53:ListQueryLoggingConfigs"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "route53resolver:ListFirewallRuleGroupAssociations",
        "route53resolver:GetFirewallRuleGroup",
        "route53resolver:ListFirewallRuleGroups",
        "route53resolver:ListResolverEndpoints",
        "route53resolver:GetResolverQueryLogConfig",
        "route53resolver:ListResolverQueryLogConfigs",
        "route53resolver:ListResolverQueryLogConfigAssociations",
        "route53resolver:GetResolverEndpoint",
        "route53resolver:GetFirewallRuleGroupAssociation"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter serviceverknüpfte Rollenberechtigungen im IAM-Benutzerhandbuch.

Eine dienstbezogene Rolle für CloudWatch Application Insights erstellen

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie in der eine neue Application Insights-Anwendung erstellen AWS Management Console, erstellt CloudWatch Application Insights die serviceverknüpfte Rolle für Sie.

Wenn Sie diese servicegebundene Rolle löschen und dann erneut anlegen möchten, können Sie die Rolle in Ihrem Konto mit demselben Verfahren neu anlegen. Wenn Sie eine neue Application

Insights-Anwendung erstellen, erstellt CloudWatch Application Insights die serviceverknüpfte Rolle erneut für Sie.

Bearbeitung einer serviceverknüpften Rolle für Application Insights CloudWatch

CloudWatch In Application Insights können Sie die AWSServiceRoleForApplicationInsights dienstverknüpfte Rolle nicht bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer dienstverknüpften Rolle für Application Insights CloudWatch

Wenn Sie ein Feature oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise vermeiden Sie, dass eine ungenutzte Einheit nicht aktiv überwacht oder gewartet wird. Sie müssen jedoch alle Anwendungen in Application Insights löschen, bevor Sie die Rolle manuell löschen können.

> **Note**
>
> Wenn der CloudWatch Application Insights-Dienst die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um CloudWatch Application Insights-Ressourcen zu löschen, die von AWSServiceRoleForApplicationInsights

- Löschen Sie alle Ihre CloudWatch Application Insights-Anwendungen. Weitere Informationen finden Sie unter „Löschen Ihrer Anwendung (en)“ im CloudWatch Application Insights-Benutzerhandbuch.

So löschen Sie die serviceverknüpfte Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die AWSServiceRoleForApplicationInsights serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Leitfaden.

Unterstützte Regionen für serviceverknüpfte Rollen mit CloudWatch Application Insights

CloudWatch Application Insights unterstützt die Verwendung von dienstbezogenen Rollen in allen AWS Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter CloudWatch Application Insights-Regionen und Endpunkte.

AWS verwaltete Richtlinien für Amazon CloudWatch Application Insights

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird AWS. AWS Verwaltete Richtlinien sind so konzipiert, dass sie Berechtigungen für viele gängige Anwendungsfälle bereitstellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie allen AWS Kunden zur Verfügung stehen. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie kundenverwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinie: CloudWatchApplicationInsightsFullAccess

Sie können die `CloudWatchApplicationInsightsFullAccess`-Richtlinie an Ihre IAM-Identitäten anfügen.

Diese Richtlinie gewährt Administratorberechtigungen, die vollen Zugriff auf die Application-Insights-Funktionalität ermöglichen.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `applicationinsights` – Ermöglicht vollen Zugriff auf die Funktionalität von Application Insights.
- `iam`— Ermöglicht Application Insights, die serviceverknüpfte Rolle zu erstellen, AWSServiceRoleForApplicationInsights. Dies ist erforderlich, damit Application Insights Operationen wie die Analyse der Ressourcengruppen eines Kunden, die Erstellung von CloudFormation Stacks zur Erstellung von Alarmen für Kennzahlen und die Konfiguration des CloudWatch Agenten auf EC2-Instances ausführen kann. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für CloudWatch Application Insights.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "applicationinsights:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeVolumes",
        "rds:DescribeDBInstances",
        "rds:DescribeDBClusters",
        "sqs:ListQueues",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetHealth",
        "autoscaling:DescribeAutoScalingGroups",
        "lambda:ListFunctions",
        "dynamodb:ListTables",
        "s3:ListAllMyBuckets",
```

```
        "sns:ListTopics",
        "states:ListStateMachines",
        "apigateway:GET",
        "ecs:ListClusters",
        "ecs:DescribeTaskDefinition",
        "ecs:ListServices",
        "ecs:ListTasks",
        "eks:ListClusters",
        "eks:ListNodegroups",
        "fsx:DescribeFileSystems",
        "logs:DescribeLogGroups",
        "elasticfilesystem:DescribeFileSystems"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/aws-service-role/application-insights.amazonaws.com/
AWSServiceRoleForApplicationInsights"
      ],
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "application-insights.amazonaws.com"
        }
      }
    }
  ]
}
```

AWS verwaltete Richtlinie: CloudWatchApplicationInsightsReadOnlyAccess

Sie können die `CloudWatchApplicationInsightsReadOnlyAccess`-Richtlinie an Ihre IAM-Identitäten anfügen.

Diese Richtlinie gewährt Administratorberechtigungen, die schreibgeschützten Zugriff auf alle Application-Insights-Funktionen ermöglichen.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `applicationinsights` – Erlaubt schreibgeschützten Zugriff auf Application-Insights-Funktionen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "applicationinsights:Describe*",
                "applicationinsights:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS verwaltete Richtlinie: CloudwatchApplicationInsightsServiceLinkedRolePolicy

Sie können keine Verbindungen CloudwatchApplicationInsightsServiceLinkedRolePolicy zu Ihren IAM-Entitäten herstellen. Diese Richtlinie ist einer servicebezogenen Rolle zugeordnet, die es Application Insights ermöglicht, Kundenressourcen zu überwachen. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für CloudWatch Application Insights.

Aktualisierungen von Application Insights auf AWS verwaltete Richtlinien

Sehen Sie sich Details zu Aktualisierungen der AWS verwalteten Richtlinien für Application Insights an, seit dieser Service begonnen hat, diese Änderungen zu verfolgen. Um automatisch über Änderungen auf dieser Seite benachrichtigt zu werden, abonnieren Sie den RSS-Feed auf der Dokumentverlaufsseite von Application Insights.

Änderung	Beschreibung	Datum
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Application Insights hat neue Berechtigungen zu CloudFormation Listenstapeln hinzugefügt. Diese Berechtigungen sind erforderlich, damit Amazon CloudWatch Application Insights die im CloudFormation Stack verschachtelten AWS Ressourcen analysieren und überwachen kann.	24. April 2023
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Application Insights hat neue Berechtigungen hinzugefügt, um die Liste der Amazon-VPC- und Route-53-Ressourcen abzufragen. Diese Berechtigungen sind erforderlich, damit Amazon CloudWatch Application Insights automatisch Best-Practice-Netzwerküberwachung einrichten kann Amazon CloudWatch.	23. Januar 2023
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Application Insights hat neue Berechtigungen hinzugefügt, um die Ergebnisse des SSM-Befehlsaufrufs abzurufen. Diese Berechtigungen sind erforderlich, damit Amazon CloudWatch Application Insights Workloads, die auf	19. Dezember 2022

Änderung	Beschreibung	Datum
	Amazon EC2 EC2-Instances ausgeführt werden, automatisch erkennen und überwachen kann.	
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Application Insights hat neue Berechtigungen hinzugefügt, um die Liste der Amazon-VPC- und Route-53-Ressourcen zu beschreiben. Diese Berechtigungen sind erforderlich, damit Amazon CloudWatch Application Insights die Amazon VPC- und Route 53-Ressourcenkonfigurationen von Kunden lesen und Kunden bei der automatischen Einrichtung von Best-Practice-Netzwerküberwachungen unterstützen kann. Amazon CloudWatch	19. Dezember 2022

Änderung	Beschreibung	Datum
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Application Insights hat neue Berechtigungen hinzugefügt, um EFS-Ressourcen zu beschreiben. Diese Berechtigungen sind erforderlich, damit Amazon CloudWatch Application Insights die Kundenressourcenkonfigurationen von Amazon EFS lesen und Kunden dabei unterstützen kann, automatisch bewährte Methoden für die EFS-Überwachung einzurichten CloudWatch.	3. Oktober 2022
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Application Insights hat neue Berechtigungen hinzugefügt, um das EFS-Dateisystem zu beschreiben. Diese Berechtigungen sind erforderlich, damit Amazon CloudWatch Application Insights kontobasierte Anwendungen erstellen kann, indem alle unterstützten Ressourcen in einem Konto abgefragt werden.	3. Oktober 2022

Änderung	Beschreibung	Datum
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Application Insights hat neue Berechtigungen hinzugefügt, um Informationen über FSx-Ressourcen abzurufen. Diese Berechtigungen sind erforderlich, damit Amazon CloudWatch Application Insights Workloads überwachen kann, indem es ausreichende Informationen über die zugrunde liegenden FSx-Volumes abruft.	12. September 2022
AWS verwaltete Richtlinie: CloudWatchApplicationInsightsFullAccess – Aktualisierung auf eine bestehende Richtlinie	Application Insights hat eine neue Berechtigung hinzugefügt, um Protokoll-Gruppen zu beschreiben. Diese Berechtigungen sind für Amazon CloudWatch Application Insights erforderlich, um sicherzustellen, dass bei der Erstellung einer neuen Anwendung die richtigen Berechtigungen für die Überwachung von Protokollgruppen in einem Konto vorhanden sind.	24. Januar 2022

Änderung	Beschreibung	Datum
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Application Insights hat neue Berechtigungen zum Erstellen und Löschen von CloudWatch Protokollabonnementfiltern hinzugefügt. Diese Berechtigungen sind erforderlich, damit Amazon CloudWatch Application Insights Abonnementfilter erstellen kann, um die Protokollüberwachung von Ressourcen in konfigurierten Anwendungen zu erleichtern.	24. Januar 2022
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Für Application Insights gibt es neue Berechtigungen, um Zielgruppen und Zielzustand für Elastic Load Balancers zu beschreiben. Diese Berechtigungen sind erforderlich, damit Amazon CloudWatch Application Insights kontobasierte Anwendungen erstellen kann, indem alle unterstützten Ressourcen in einem Konto abgefragt werden.	4. November 2021

Änderung	Beschreibung	Datum
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Für Application Insights gibt es neue Berechtigungen zum Ausführen des `AmazonCloudWatch-ManageAgent` -SSM-Dokuments auf Amazon-EC2-Instances. Diese Berechtigungen sind erforderlich, damit Amazon CloudWatch Application Insights die von Application Insights erstellten CloudWatch Agentenkonfigurationsdateien bereinigen kann.	30. September 2021

Änderung	Beschreibung	Datum
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Für Application Insights gibt es neue Berechtigungen, um die kontobasierte Anwendungsüberwachung zu unterstützen und alle unterstützten Ressourcen in Ihrem Konto zu integrieren und zu überwachen. Diese Berechtigungen sind erforderlich, damit Amazon CloudWatch Application Insights Ressourcen abfragen, taggen und Gruppen für diese Ressourcen erstellen kann. Für Application Insights gibt es neue Berechtigungen, um die Überwachung von SNS-Themen zu unterstützen. Diese Berechtigungen sind erforderlich, damit Amazon CloudWatch Application Insights Metadaten aus SNS-Ressourcen sammeln und die Überwachung für SNS-Themen konfigurieren kann.	15. September 2021

Änderung	Beschreibung	Datum
AWS verwaltete Richtlinie: CloudWatchApplicationInsightsFullAccess – Aktualisierung auf eine bestehende Richtlinie	Für Application Insights gibt es neue Berechtigungen, um unterstützte Ressourcen zu beschreiben und aufzulisten. Diese Berechtigungen sind erforderlich, damit Amazon CloudWatch Application Insights kontobasierte Anwendungen erstellen kann, indem alle unterstützten Ressourcen in einem Konto abgefragt werden.	15. September 2021
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Application Insights hat neue Berechtigungen hinzugefügt, um FSx-Ressourcen zu beschreiben. Diese Berechtigungen sind erforderlich, damit Amazon CloudWatch Application Insights die FSx-Ressourcenkonfigurationen von Kunden lesen und Kunden bei der automatischen Einrichtung der Best-Practice-FSx-Überwachung unterstützen kann. CloudWatch	31. August 2021

Änderung	Beschreibung	Datum
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Application Insights hat neue Berechtigungen hinzugefügt, um ECS- und EKS-Serviceressourcen zu beschreiben und aufzulisten. Diese Berechtigung ist erforderlich, damit Amazon CloudWatch Application Insights die Konfiguration der Kundencontainerressourcen lesen und Kunden dabei helfen kann, die Container-Überwachung mit bewährten Methoden automatisch einzurichten CloudWatch.	18. Mai 2021
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Application Insights hat neue Berechtigungen hinzugefügt, die es OpsCenter ermöglichen, OpsItems mithilfe der `ssm:AddTagsToResource` Aktion Tags für Ressourcen mit dem `opsitem` Ressourcentyp zu markieren. Diese Berechtigung ist erforderlich von OpsCenter. Amazon CloudWatch Application Insights erstellt, OpsItems damit der Kunde Probleme mithilfe von AWS SSM OpsCenter lösen kann.	13. April 2021

Änderung	Beschreibung	Datum
Application Insights hat mit der Verfolgung von Änderungen begonnen	Application Insights hat damit begonnen, Änderungen an den AWS verwalteten Richtlinien nachzuverfolgen.	13. April 2021

Referenz zu CloudWatch Amazon-Berechtigungen

In der folgenden Tabelle sind die einzelnen CloudWatch API-Operationen und die entsprechenden Aktionen aufgeführt, für die Sie Berechtigungen zur Ausführung der Aktion erteilen können. Die Aktionen geben Sie im Feld `Action` und ein Platzhalterzeichen (*) als Wert für die Ressource im Feld `Resource` der Richtlinie an.

Sie können in Ihren CloudWatch Richtlinien AWS allgemeine Bedingungsschlüssel verwenden, um Bedingungen auszudrücken. Eine vollständige Liste der Schlüssel für alle AWS Benutzer finden Sie unter AWS Globale Schlüssel und IAM-Bedingungskontextschlüssel im IAM-Benutzerhandbuch.

> **Note**
>
> Um eine Aktion anzugeben, verwenden Sie das Präfix `cloudwatch:` gefolgt vom Namen der API-Operation. Beispiel: `cloudwatch:GetMetricData`, `cloudwatch:ListMetrics` oder `cloudwatch:*` (für alle CloudWatch-Aktionen).

Themen

- CloudWatch API-Operationen und erforderliche Berechtigungen für Aktionen
- CloudWatch API-Operationen von Contributor Insights und erforderliche Berechtigungen für Aktionen
- CloudWatch API-Operationen für Ereignisse und erforderliche Berechtigungen für Aktionen
- CloudWatch Protokolliert API-Operationen und erforderliche Berechtigungen für Aktionen
- Amazon-EC2-API-Operationen und erforderliche Berechtigungen für Aktionen
- API-Operationen von Amazon EC2 Auto Scaling und erforderliche Berechtigungen für Aktionen

CloudWatch API-Operationen und erforderliche Berechtigungen für Aktionen

CloudWatch API-Operationen	Erforderliche Berechtigungen (API-Aktionen)
DeleteAlarms	`cloudwatch:DeleteAlarms` Erforderlich zum Löschen eines Alarms.
DeleteDashboards	`cloudwatch:DeleteDashboards` Erforderlich zum Löschen eines Dashboards
DeleteMetricStream	`cloudwatch:DeleteMetricStream` Erforderlich zum Löschen eines Metrik-Streams.
DescribeAlarmHistory	`cloudwatch:DescribeAlarmHistory` Erforderlich zum Anzeigen des Alarmverlauf. Um Informationen zu zusammengesetzten Alarmen abzurufen, muss Ihre `cloudwatch:DescribeAlarmHistory` -Berechtigung einen *-Bereich haben. Sie können keine Informationen zu zusammengesetzten Alarmen zurückgeben, wenn Ihre `cloudwatch:DescribeAlarmHistory` -Berechtigung einen engeren Geltungsbereich hat.
DescribeAlarms	`cloudwatch:DescribeAlarms` Erforderlich zum Abrufen von Informationen zu einem Alarm. Um Informationen zu zusammengesetzten Alarmen abzurufen, muss Ihre `cloudwatc`

CloudWatch API-Operationen	Erforderliche Berechtigungen (API-Aktionen)
	`h:DescribeAlarms` -Berechtigung einen *-Bereich haben. Sie können keine Informationen zu zusammengesetzten Alarmen zurückgeben, wenn Ihre `cloudwatch:DescribeAlarms` -Berechtigung einen engeren Geltungsbereich hat.
DescribeAlarmsForMetric	`cloudwatch:DescribeAlarmsForMetric` Erforderlich zum Anzeigen von Alarmen für eine Metrik.
DisableAlarmActions	`cloudwatch:DisableAlarmActions` Erforderlich zum Deaktivieren einer Alarmaktion.
EnableAlarmActions	`cloudwatch:EnableAlarmActions` Erforderlich zum Aktivieren einer Alarmaktion.
GetDashboard	`cloudwatch:GetDashboard` Erforderlich zum Anzeigen von Daten über vorhandene Dashboards.
GetMetricData	`cloudwatch:GetMetricData` Erforderlich, um metrische Daten in der CloudWatch Konsole grafisch darzustellen, große Stapel metrischer Daten abzurufen und metrische Berechnungen mit diesen Daten durchzuführen.

CloudWatch API-Operationen	Erforderliche Berechtigungen (API-Aktionen)
GetMetricStatistics	`cloudwatch:GetMetricStatistics` Erforderlich, um Diagramme in anderen Teilen der CloudWatch Konsole und in Dashboard-Widgets anzuzeigen.
GetMetricStream	`cloudwatch:GetMetricStream` Erforderlich, um Informationen zu einem Metrik-Stream anzuzeigen.
GetMetricWidgetImage	`cloudwatch:GetMetricWidgetImage` Erforderlich, um ein Snapshot-Diagramm mit einer oder mehreren CloudWatch Metriken als Bitmap-Bild abzurufen.
ListDashboards	`cloudwatch:ListDashboards` Erforderlich, um die Liste der CloudWatch Dashboards in Ihrem Konto anzuzeigen.
ListMetrics	`cloudwatch:ListMetrics` Erforderlich, um Metriknamen in der CloudWatch Konsole und in der CLI anzuzeigen oder zu suchen. Erforderlich zum Auswählen von Metriken auf Dashboard-Widgets.
ListMetricStreams	`cloudwatch:ListMetricStreams` Erforderlich zum Anzeigen oder Durchsuchen der Liste der Metrik-Streams im Konto.

CloudWatch API-Operationen	Erforderliche Berechtigungen (API-Aktionen)
PutCompositeAlarm	`cloudwatch:PutCompositeAlarm` Erforderlich, um einen zusammengesetzten Alarm zu erstellen. Um einen zusammengesetzten Alarm zu erstellen, muss Ihre `cloudwatch:PutCompositeAlarm` -Berechtigung einen *-Bereich haben. Sie können keine Informationen zu zusammengesetzten Alarmen zurückgeben, wenn Ihre `cloudwatch:PutCompositeAlarm` -Berechtigung einen engeren Geltungsbereich hat.
PutDashboard	`cloudwatch:PutDashboard` Erforderlich zum Erstellen eines Dashboard oder zum Aktualisieren eines vorhandenen Dashboards.
PutMetricAlarm	`cloudwatch:PutMetricAlarm` Erforderlich zum Erstellen und Aktualisieren eines Alarms.
PutMetricData	`cloudwatch:PutMetricData` Erforderlich zum Erstellen einer Metrik.
PutMetricStream	`cloudwatch:PutMetricStream` Erforderlich zum Erstellen eines Metrik-Streams.

CloudWatch API-Operationen	Erforderliche Berechtigungen (API-Aktionen)
SetAlarmState	`cloudwatch:SetAlarmState` Erforderlich zum manuellen Einrichten des Status eines Alarms.
StartMetricStreams	`cloudwatch:StartMetricStreams` Erforderlich, um den Fluss von Metriken in einem Metrik-Stream zu starten.
StopMetricStreams	`cloudwatch:StopMetricStreams` Erforderlich, um den Fluss von Metriken in einem Metrik-Stream vorübergehend zu stoppen.
TagResource	`cloudwatch:TagResource` Erforderlich, um Tags zu CloudWatch Ressourcen wie Alarmen und Contributor Insights-Regeln hinzuzufügen oder zu aktualisieren.
UntagResource	`cloudwatch:UntagResource` Erforderlich, um Tags aus CloudWatch Ressourcen zu entfernen.

CloudWatch API-Operationen von Contributor Insights und erforderliche Berechtigungen für Aktionen

> **Important**
>
> Wenn Sie einem Benutzer die `cloudwatch:PutInsightRule` Berechtigung erteilen, kann dieser Benutzer standardmäßig eine Regel erstellen, die jede Protokollgruppe in CloudWatch Logs auswertet. Sie können IAM-Richtlinienbedingungen hinzufügen, die diese Berechtigungen für einen Benutzer einschränken, um bestimmte Protokollgruppen einzuschließen und auszuschließen. Weitere Informationen finden Sie unter Verwenden von Bedingungsschlüsseln, um den Zugriff von Contributor-Insights-Benutzern auf Protokollgruppen einzuschränken.

CloudWatch API-Operationen von Contributor Insights	Erforderliche Berechtigungen (API-Aktionen)
DeleteInsightRules	`cloudwatch:DeleteInsightRules` Erforderlich, um Contributor-Insights-Regeln zu löschen
DescribeInsightRules	`cloudwatch:DescribeInsightRules` Erforderlich für die Anzeige der Contributor-Insights-Regeln in Ihrem Konto.
EnableInsightRules	`cloudwatch:EnableInsightRules` Erforderlich, um Contributor-Insights-Regeln zu aktivieren.
GetInsightRuleReport	`cloudwatch:GetInsightRuleReport`

CloudWatch API-Operationen von Contributor Insights	Erforderliche Berechtigungen (API-Aktionen)
	Erforderlich zum Abrufen von Zeitreihendaten und anderen Statistiken, die von Contributor-Insights-Regeln erfasst wurden.
PutInsightRule	`cloudwatch:PutInsightRule` Erforderlich, um Contributor-Insights-Regeln zu erstellen. Siehe den Wichtig-Hinweis am Anfang dieser Tabelle.

CloudWatch API-Operationen für Ereignisse und erforderliche Berechtigungen für Aktionen

CloudWatch API-Operationen für Ereignisse	Erforderliche Berechtigungen (API-Aktionen)
DeleteRule	`events:DeleteRule` Erforderlich zum Löschen einer Regel.
DescribeRule	`events:DescribeRule` Erforderlich zum Auflisten der Details einer Regel.
DisableRule	`events:DisableRule` Erforderlich zum Deaktivieren einer Regel.
EnableRule	`events:EnableRule` Erforderlich zum Aktivieren einer Regel.

CloudWatch API-Operationen für Ereignisse	Erforderliche Berechtigungen (API-Aktionen)
ListRuleNamesByTarget	`events:ListRuleNamesByTarget` Erforderlich zum Auflisten von Regeln, die mit einem Ziel verknüpft sind.
ListRules	`events:ListRules` Erforderlich zum Auflisten aller Regeln in Ihrem Konto.
ListTargetsByRule	`events:ListTargetsByRule` Erforderlich zum Auflisten aller Ziele im Zusammenhang mit einer Regel.
PutEvents	`events:PutEvents` Erforderlich zum Hinzufügen von benutzers pezifischen Ereignissen, die Regeln zugeordne t werden können.
PutRule	`events:PutRule` Erforderlich zum Erstellen oder Aktualisieren einer Regel.
PutTargets	`events:PutTargets` Erforderlich zum Hinzufügen von Zielen zu einer Regel.
RemoveTargets	`events:RemoveTargets` Erforderlich zum Entfernen eines Ziels aus einer Regel.

CloudWatch API-Operationen für Ereignisse	Erforderliche Berechtigungen (API-Aktionen)
TestEventPattern	`events:TestEventPattern` Erforderlich zum Testen eines Ereignismusters für ein bestimmtes Ereignis.

CloudWatch Protokolliert API-Operationen und erforderliche Berechtigungen für Aktionen

CloudWatch Protokolliert API-Operationen	Erforderliche Berechtigungen (API-Aktionen)
CancelExportTask	`logs:CancelExportTask` Erforderlich zum Abbrechen einer ausstehenden oder laufenden Exportaufgabe.
CreateExportTask	`logs:CreateExportTask` Erforderlich zum Exportieren von Daten aus einer Protokollgruppe in einen Amazon-S3-Bucket.
CreateLogGroup	`logs:CreateLogGroup` Erforderlich zum Erstellen einer neuen Protokoll gruppe.
CreateLogStream	`logs:CreateLogStream` Erforderlich zum Erstellen eines neuen Protokoll-Stream in eine Protokollgruppe.
DeleteDestination	`logs:DeleteDestination`

CloudWatch Protokolliert API-Operationen	Erforderliche Berechtigungen (API-Aktionen)
	Erforderlich zum Löschen eines Protokollziels und Aktivieren von Abonnementfiltern für das Ziel.
DeleteLogGroup	`logs:DeleteLogGroup` Erforderlich zum Löschen einer Protokoll gruppe und der jeweiligen archivierten Protokoll ereignisse.
DeleteLogStream	`logs:DeleteLogStream` Erforderlich zum Löschen eines Protokoll-Stream und der jeweiligen archivierten Protokoll ereignisse.
DeleteMetricFilter	`logs:DeleteMetricFilter` Erforderlich zum Löschen eines Metrikfilters für eine Protokollgruppe.
DeleteQueryDefinition	`logs:DeleteQueryDefinition` Erforderlich, um eine gespeicherte Abfragede finition in CloudWatch Logs Insights zu löschen.
DeleteResourcePolicy	`logs:DeleteResourcePolicy` Erforderlich, um eine CloudWatch Logs-Ress ourcenrichtlinie zu löschen.

CloudWatch Protokolliert API-Operationen	Erforderliche Berechtigungen (API-Aktionen)
DeleteRetentionPolicy	`logs:DeleteRetentionPolicy` Erforderlich zum Löschen der Aufbewahr ungsrichtlinie einer Protokollgruppe.
DeleteSubscriptionFilter	`logs:DeleteSubscriptionFilter` Erforderlich zum Löschen des Abonnemen tfilters für eine Protokollgruppe.
DescribeDestinations	`logs:DescribeDestinations` Erforderlich zum Anzeigen aller Ziele für ein Konto.
DescribeExportTasks	`logs:DescribeExportTasks` Erforderlich zum Anzeigen aller Exportauf gaben für das Konto.
DescribeLogGroups	`logs:DescribeLogGroups` Erforderlich zum Anzeigen aller Protokoll gruppen für ein Konto.
DescribeLogStreams	`logs:DescribeLogStreams` Erforderlich zum Anzeigen aller Protokoll-Streams für eine Protokollgruppe.
DescribeMetricFilters	`logs:DescribeMetricFilters` Erforderlich zum Anzeigen aller Metriken für eine Protokollgruppe.

CloudWatch Protokolliert API-Operationen	Erforderliche Berechtigungen (API-Aktionen)
DescribeQueryDefinitions	`logs:DescribeQueryDefinitions` Erforderlich, um die Liste der gespeicherten Abfragedefinitionen in CloudWatch Logs Insights zu sehen.
DescribeQueries	`logs:DescribeQueries` Erforderlich, um die Liste der CloudWatch Logs Insights-Abfragen zu sehen, die geplant sind, ausgeführt werden oder kürzlich ausgeführt wurden.
DescribeResourcePolicies	`logs:DescribeResourcePolicies` Erforderlich, um eine Liste der CloudWatch Logs-Ressourcenrichtlinien anzuzeigen.
DescribeSubscriptionFilters	`logs:DescribeSubscriptionFilters` Erforderlich zum Anzeigen aller Abonnemen tfilter für eine Protokollgruppe.
FilterLogEvents	`logs:FilterLogEvents` Erforderlich zum Sortieren von Protokoll ereignissen nach Gruppenfiltermuster.
GetLogEvents	`logs:GetLogEvents` Erforderlich zum Abrufen von Protokollereigniss en aus einem Protokoll-Stream.

CloudWatch Protokolliert API-Operationen	Erforderliche Berechtigungen (API-Aktionen)
GetLogGroupFields	`logs:GetLogGroupFields` Erforderlich, um die Liste der Felder abzurufen , die in den Protokollereignissen in einer Protokollgruppe enthalten sind.
GetLogRecord	`logs:GetLogRecord` Erforderlich, um die Details aus einem einzelnen Protokollereignis abzurufen.
GetQueryResults	`logs:GetQueryResults` Erforderlich, um die Ergebnisse von CloudWatch Logs Insights-Abfragen abzurufen.
ListTagsLogGroup	`logs:ListTagsLogGroup` Erforderlich zum Auflisten der mit einer Protokollgruppe verbundenen Tags.
PutDestination	`logs:PutDestination` Erforderlich zum Erstellen oder Aktualisieren eines Ziel-Protokoll-Streams (z. B. ein Kinesis-Stream).
PutDestinationPolicy	`logs:PutDestinationPolicy` Erforderlich zum Erstellen oder Aktualisieren einer Zugriffsrichtlinie im Zusammenhang mit einem vorhandenen Protokollziel.

CloudWatch Protokolliert API-Operationen	Erforderliche Berechtigungen (API-Aktionen)
PutLogEvents	`logs:PutLogEvents` Erforderlich für den Upload eines Batches von Protokollereignissen in einen Protokoll-Stream.
PutMetricFilter	`logs:PutMetricFilter` Erforderlich zum Erstellen oder Aktualisieren eines Metrikfilters, der einer Protokollgruppe zugeordnet wird.
PutQueryDefinition	`logs:PutQueryDefinition` Erforderlich, um eine Abfrage in CloudWatch Logs Insights zu speichern.
PutResourcePolicy	`logs:PutResourcePolicy` Erforderlich, um eine CloudWatch Logs-Ress ourcenrichtlinie zu erstellen.
PutRetentionPolicy	`logs:PutRetentionPolicy` Erforderlich zum Festlegen der Anzahl der Tage, die Protokollereignisse (Aufbewahrung) in einer Protokollgruppe aufbewahrt werden sollen.
PutSubscriptionFilter	`logs:PutSubscriptionFilter` Erforderlich zum Erstellen oder Aktualisieren eines Abonnementfilters, der einer Protokoll gruppe zugeordnet wird.

CloudWatch Protokolliert API-Operationen	Erforderliche Berechtigungen (API-Aktionen)
StartQuery	`logs:StartQuery` Erforderlich, um CloudWatch Logs Insights-Abfragen zu starten.
StopQuery	`logs:StopQuery` Erforderlich, um eine laufende CloudWatch Logs Insights-Abfrage zu beenden.
TagLogGroup	`logs:TagLogGroup` Erforderlich zum Hinzufügen oder Aktualisieren von Protokollgruppen-Tags.
TestMetricFilter	`logs:TestMetricFilter` Erforderlich zum Testen eines Filtermusters anhand einer Stichprobe von Protokollereignis-Nachrichten.

Amazon-EC2-API-Operationen und erforderliche Berechtigungen für Aktionen

Amazon-EC2-API-Operationen	Erforderliche Berechtigungen (API-Aktionen)
DescribeInstanceStatus	`ec2:DescribeInstanceStatus` Erforderlich zum Anzeigen von Details des EC2 Instance-Status.
DescribeInstances	`ec2:DescribeInstances`

Amazon-EC2-API-Operationen	Erforderliche Berechtigungen (API-Aktionen)
	Erforderlich zum Anzeigen von EC2 Instance-Details.
RebootInstances	`ec2:RebootInstances` Erforderlich zum Neustarten einer EC2 Instance.
StopInstances	`ec2:StopInstances` Erforderlich zum Anhalten einer EC2 Instance.
TerminateInstances	`ec2:TerminateInstances` Erforderlich zum Beenden einer EC2 Instance.

API-Operationen von Amazon EC2 Auto Scaling und erforderliche Berechtigungen für Aktionen

Amazon-EC2-Auto-Scaling-API-Operationen	Erforderliche Berechtigungen (API-Aktionen)
Skalierung	`autoscaling:Scaling` Erforderlich zum Skalieren einer Auto-Scaling-Gruppe.
Auslöser	`autoscaling:Trigger` Erforderlich zum Auslösen einer Auto-Scaling-Aktion.

Konformitätsvalidierung für Amazon CloudWatch

Externe Prüfer bewerten die Sicherheit und Konformität von Amazon im CloudWatch Rahmen mehrerer AWS Compliance-Programme. Hierzu zählen unter anderem SOC, PCI, FedRAMP und HIPAA.

Eine Liste der AWS Services im Rahmen bestimmter Compliance-Programme finden Sie unter AWS Services im Umfang nach Compliance-Programmen AWS . Allgemeine Informationen finden Sie unter AWS -Compliance-Programme.

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

Ihre Compliance-Verantwortung bei der Nutzung von Amazon CloudWatch hängt von der Sensibilität Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- Schnellstartanleitungen für Sicherheit und Compliance – In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Bereitstellung von sicherheits- und konformitätsorientierten Basisumgebungen auf AWS angegeben.
- Whitepaper „Architecting for HIPAA Security and Compliance“ — In diesem Whitepaper wird beschrieben, wie Unternehmen HIPAA-konforme Anwendungen erstellen können AWS .
- AWS Compliance-Ressourcen — Diese Sammlung von Arbeitsmappen und Leitfäden kann auf Ihre Branche und Ihren Standort zutreffen.
- Bewertung von Ressourcen anhand von Regeln im AWS Config Entwicklerhandbuch — AWS Config; bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub— Dieser AWS Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus und hilft Ihnen AWS , die Einhaltung der Sicherheitsstandards und Best Practices der Branche zu überprüfen.

Resilienz bei Amazon CloudWatch

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones. Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die über hoch redundante

Netzwerke mit niedriger Latenz und hohen Durchsätzen verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Infrastruktursicherheit bei Amazon CloudWatch

Als verwalteter Service CloudWatch ist Amazon durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter AWS Cloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe für den Zugriff CloudWatch über das Netzwerk. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Netzwerkisolierung

Eine Virtual Private Cloud (VPC) ist ein virtuelles Netzwerk in Ihrem eigenen logisch isolierten Bereich in der Cloud von Amazon Web Services. Ein Subnetz ist ein Bereich von IP-Adressen in einer VPC. Sie können eine Vielzahl von AWS -Ressourcen in den Subnetzen Ihrer VPCs bereitstellen. Sie können beispielsweise Amazon-EC2-Instances, EMR-Cluster und DynamoDB-Tabellen in Subnetzen bereitstellen. Weitere Informationen finden Sie im Amazon VPC-Benutzerhandbuch.

Um die Kommunikation mit Ressourcen in einer VPC CloudWatch zu ermöglichen, ohne das öffentliche Internet nutzen zu müssen, verwenden Sie AWS PrivateLink. Weitere Informationen finden Sie unter Verwendung von CloudWatch und CloudWatch Synthetics mit VPC-Endpunkten mit Schnittstelle.

Ein privates Subnetz ist ein Subnetz ohne Standardroute zum öffentlichen Internet. Die Bereitstellung einer AWS Ressource in einem privaten Subnetz hindert Amazon CloudWatch nicht daran, integrierte Metriken von der Ressource zu sammeln.

Wenn Sie benutzerdefinierte Messwerte von einer AWS Ressource in einem privaten Subnetz veröffentlichen müssen, können Sie dies mithilfe eines Proxyservers tun. Der Proxyserver leitet diese HTTPS-Anfragen an die öffentlichen API-Endpunkte für weiter. CloudWatch

AWS Security Hub

Überwachen Sie Ihre Nutzung von CloudWatch in Bezug auf bewährte Sicherheitsmethoden mithilfe von AWS Security Hub. Security Hub verwendet Sicherheitskontrollen für die Bewertung von Ressourcenkonfigurationen und Sicherheitsstandards, um Sie bei der Einhaltung verschiedener Compliance-Frameworks zu unterstützen. Weitere Informationen zur Verwendung von Security Hub zur Bewertung von CloudWatch Ressourcen finden Sie unter Amazon CloudWatch Controls im AWS Security Hub Hub-Benutzerhandbuch.

Verwendung von CloudWatch und CloudWatch Synthetics mit VPC-Endpunkten mit Schnittstelle

Wenn Sie Amazon Virtual Private Cloud (Amazon VPC) zum Hosten Ihrer AWS Ressourcen verwenden, können Sie eine private Verbindung zwischen Ihrer VPC und CloudWatch Synthetics CloudWatch herstellen. Sie können diese Verbindungen verwenden, um CloudWatch Synthetics die Kommunikation mit Ihren Ressourcen auf Ihrer VPC zu ermöglichen CloudWatch , ohne das öffentliche Internet nutzen zu müssen.

Amazon VPC ist ein AWS Service, mit dem Sie AWS Ressourcen in einem von Ihnen definierten virtuellen Netzwerk starten können. Mit einer VPC haben Sie die Kontrolle über Ihre Netzwerkeinstellungen, wie IP-Adressbereich, Subnetze, Routing-Tabellen und Netzwerk-Gateways. Um Ihre VPC mit CloudWatch Synthetics zu CloudWatch verbinden, definieren Sie einen VPC-Schnittstellen-Endpunkt, um Ihre VPC mit Diensten zu verbinden. AWS Der Endpunkt bietet zuverlässige, skalierbare Konnektivität zu CloudWatch oder CloudWatch Synthetics, ohne dass ein

Internet-Gateway, eine NAT-Instance (Network Address Translation) oder eine VPN-Verbindung erforderlich ist. Weitere Informationen finden Sie unter Was ist Amazon VPC im Benutzerhandbuch zu Amazon VPC.

Schnittstelle, auf der VPC-Endpunkte basieren AWS PrivateLink, eine AWS Technologie, die private Kommunikation zwischen AWS Diensten über eine elastic network interface mit privaten IP-Adressen ermöglicht. Weitere Informationen finden Sie im Blogbeitrag New — AWS PrivateLink for AWS Services.

Die folgenden Schritte sind für Benutzer von Amazon VPC vorgesehen. Weitere Informationen finden Sie unter Erste Schritte im Amazon VPC Benutzerhandbuch.

CloudWatch VPC-Endpunkt

CloudWatch unterstützt derzeit VPC-Endpunkte in den folgenden Regionen: AWS

- US East (Ohio)
- USA Ost (Nord-Virginia)
- USA West (Nordkalifornien)
- USA West (Oregon)
- Asien-Pazifik (Hongkong)
- Asien-Pazifik (Mumbai)
- Asia Pacific (Seoul)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)
- Canada (Central)
- Europe (Frankfurt)
- Europa (Irland)
- Europe (London)
- Europa (Paris)
- Naher Osten (VAE)
- Südamerika (São Paulo)
- AWS GovCloud (US-Ost)
- AWS GovCloud (US-West)

Erstellen eines VPC-Endpunkts für CloudWatch

Um mit der Verwendung CloudWatch mit Ihrer VPC zu beginnen, erstellen Sie einen VPC-Schnittstellen-Endpunkt für. CloudWatch Der zu wählende Service-Name lautet `com.amazonaws.region.monitoring`. Weitere Informationen finden Sie unter Erstellen eines Schnittstellenendpunkts im Amazon VPC Leitfaden.

Sie müssen die Einstellungen für nicht ändern. CloudWatch CloudWatch ruft andere AWS Dienste auf, die entweder öffentliche Endpunkte oder VPC-Endpunkte mit privaten Schnittstellen verwenden, je nachdem, welche verwendet werden. Wenn Sie beispielsweise einen VPC-Schnittstellen-Endpunkt für CloudWatch erstellen und bereits Metriken CloudWatch von Ressourcen auf Ihrer VPC zu Ihnen fließen, beginnen diese Metriken standardmäßig, über den Schnittstellen-VPC-Endpunkt zu fließen.

Steuern des Zugriffs auf Ihren CloudWatch VPC-Endpunkt

Eine VPC-Endpunktrichtlinie ist eine IAM-Ressourcenrichtlinie, die Sie einem Endpunkt beim Erstellen oder Ändern des Endpunkts zuordnen. Wenn Sie einem Endpunkt beim Erstellen keine Richtlinie zuordnen, ordnet Amazon VPC ihm eine Standardrichtlinie mit Vollzugriff auf den Service zu. -Benutzerrichtlinien oder servicespezifische Richtlinien werden durch Endpunktrichtlinien nicht überschrieben oder ersetzt. Endpunktrichtlinien steuern unabhängig vom Endpunkt den Zugriff auf den angegebenen Service.

Endpunktrichtlinien müssen im JSON-Format erstellt werden.

Weitere Informationen finden Sie unter Steuerung des Zugriffs auf Services mit VPC-Endpunkten im Amazon VPC User Guide.

Im Folgenden finden Sie ein Beispiel für eine Endpunktrichtlinie für. CloudWatch Diese Richtlinie ermöglicht Benutzern, die eine Verbindung CloudWatch über die VPC herstellen, das Senden von Metrikdaten an CloudWatch und verhindert, dass sie andere CloudWatch Aktionen ausführen.

```
{
  "Statement": [
    {
      "Sid": "PutOnly",
      "Principal": "*",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Effect": "Allow",
      "Resource": "*"
```

```
    }
  ]
}
```

So bearbeiten Sie die VPC-Endpunktrichtlinie für CloudWatch

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Endpunkte aus.
3. Wenn Sie den Endpunkt für noch nicht erstellt haben CloudWatch, wählen Sie Create Endpoint aus. Wählen Sie com.amazonaws.***Region***.monitoring und danach Create endpoint (Endpunkt erstellen) aus.
4. Wählen Sie den Endpunkt com.amazonaws.***region***.monitoring und dann die Registerkarte Policy (Richtlinie) aus.
5. Klicken Sie auf Edit Policy (Richtlinie bearbeiten) und nehmen Sie Ihre Änderungen vor.

CloudWatch VPC-Endpunkt von Synthetics

CloudWatch Synthetics unterstützt derzeit VPC-Endpunkte in den folgenden Regionen: AWS

- US East (Ohio)
- USA Ost (Nord-Virginia)
- USA West (Nordkalifornien)
- USA West (Oregon)
- Asien-Pazifik (Hongkong)
- Asien-Pazifik (Mumbai)
- Asia Pacific (Seoul)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)
- Canada (Central)
- Europe (Frankfurt)
- Europa (Irland)
- Europe (London)
- Europe (Paris)

- Südamerika (São Paulo)

Erstellen eines VPC-Endpunkts für Synthetics CloudWatch

Um mit der Verwendung von CloudWatch Synthetics mit Ihrer VPC zu beginnen, erstellen Sie einen VPC-Schnittstellen-Endpunkt für Synthetics. CloudWatch Der zu wählende Service-Name lautet `com.amazonaws.region.synthetics`. Weitere Informationen finden Sie unter Erstellen eines Schnittstellenendpunkts im Amazon VPC Leitfaden.

Sie müssen die Einstellungen für CloudWatch Synthetics nicht ändern. CloudWatch Synthetics kommuniziert mit anderen AWS Diensten entweder über öffentliche Endpunkte oder über VPC-Endpunkte mit privaten Schnittstellen, je nachdem, welche verwendet werden. Wenn Sie beispielsweise einen VPC-Schnittstellen-Endpunkt für CloudWatch Synthetics erstellen und bereits über einen Schnittstellenendpunkt für Amazon S3 verfügen, beginnt CloudWatch Synthetics standardmäßig mit der Kommunikation mit Amazon S3 über den Schnittstellen-VPC-Endpunkt.

Steuern des Zugriffs auf Ihren CloudWatch Synthetics VPC-Endpunkt

Eine VPC-Endpunktrichtlinie ist eine IAM-Ressourcenrichtlinie, die Sie einem Endpunkt beim Erstellen oder Ändern des Endpunkts zuordnen. Wenn Sie einem Endpunkt beim Erstellen keine Richtlinie zuordnen, wird ihm eine Standardrichtlinie mit Vollzugriff auf den Service zugeordnet. -Benutzerrichtlinien oder servicespezifische Richtlinien werden durch Endpunktrichtlinien nicht überschrieben oder ersetzt. Endpunktrichtlinien steuern unabhängig vom Endpunkt den Zugriff auf den angegebenen Service.

Endpunkt-Richtlinien wirken sich auf Canarys aus, die privat von der VPC verwaltet werden. Sie werden nicht für Canarys benötigt, die in privaten Subnetzen ausgeführt werden.

Endpunktrichtlinien müssen im JSON-Format erstellt werden.

Weitere Informationen finden Sie unter Steuerung des Zugriffs auf Services mit VPC-Endpunkten im Amazon VPC User Guide.

Das Folgende ist ein Beispiel für eine Endpunktrichtlinie für CloudWatch Synthetics. Diese Richtlinie ermöglicht es Benutzern, die über die VPC eine Verbindung zu CloudWatch Synthetics herstellen, Informationen über Canaries und ihre Läufe einzusehen, jedoch keine Canaries zu erstellen, zu ändern oder zu löschen.

```
{
    "Statement": [
```

```
        {
            "Action": [
                "synthetics:DescribeCanaries",
                "synthetics:GetCanaryRuns"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

So bearbeiten Sie die VPC-Endpunktrichtlinie für Synthetics CloudWatch

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Endpunkte aus.
3. Wenn Sie den Endpunkt für CloudWatch Synthetics noch nicht erstellt haben, wählen Sie Create Endpoint. Wählen Sie com.amazonaws.***region***.synthetics und dann die Option Create endpoint (Endpunkt erstellen) aus.
4. Wählen Sie com.amazonaws.***region***.synthetics endpoint und dann die Registerkarte Policy (Richtlinie) aus.
5. Klicken Sie auf Edit Policy (Richtlinie bearbeiten) und nehmen Sie Ihre Änderungen vor.

Sicherheitsüberlegungen für Synthetics-Canaries

In den folgenden Abschnitten werden Sicherheitsprobleme erläutert, die Sie beim Erstellen und Ausführen von Canaries in Synthetics berücksichtigen sollten.

Verwenden sicherer Verbindungen

Da Canary-Code und die Ergebnisse von Canary-Testläufen vertrauliche Informationen enthalten können, sollten Sie Ihre Canary-Verbindung mit Endpunkten nicht über unverschlüsselte Verbindungen herstellen. Verwenden Sie immer verschlüsselte Verbindungen, z. B. solche, die mit `https://` beginnen.

Erwägungen zur Canary-Benennung

Der Amazon-Ressourcenname (ARN) eines Canary ist im User-Agent-Header als Teil ausgehender Aufrufe enthalten, die vom Puppeteer-gesteuerten Chromium-Browser getätigt werden, der Teil der

Synthetics-Wrapper-Bibliothek ist. CloudWatch Dies hilft dabei, den Canary-Verkehr von CloudWatch Synthetics zu identifizieren und ihn mit den Kanaren in Verbindung zu bringen, die Anrufe tätigen.

Der Canary-ARN enthält den Canary-Namen. Wählen Sie Canary-Namen aus, die keine proprietären Informationen enthalten.

Stellen Sie außerdem sicher, dass Ihre Canaries nur auf Websites und Endpunkte zeigen, die Sie steuern.

Secrets und vertrauliche Informationen im Canary-Code

Wenn du deinen Canary-Code mithilfe einer Zip-Datei direkt an den Canary weitergibst, kann der Inhalt des Skripts in AWS CloudTrail Logs eingesehen werden.

Wenn Sie vertrauliche Informationen oder Secrets (wie Zugriffsschlüssel oder Datenbankanmeldeinformationen) in einem Canary-Skript haben, wird dringend empfohlen, dass Sie das Skript als versioniertes Objekt in Amazon S3 speichern und den Amazon-S3-Speicherort im Canary-Code übergeben, anstatt den Canary-Code per Zip-Datei zu übergeben.

Wenn Sie eine Zip-Datei zur Übergabe des Canary-Skripts verwenden, wird dringend empfohlen, keine Secrets oder vertrauliche Informationen in Ihren Canary-Quellcode aufzunehmen. Weitere Informationen darüber, wie du deine Geheimnisse schützen kannst AWS Secrets Manager , findest du unter Was ist AWS Secrets Manager? .

Überlegungen zu Berechtigungen

Wir empfehlen, den Zugriff auf Ressourcen zu beschränken, die von CloudWatch Synthetics erstellt oder verwendet werden. Verwenden Sie strenge Berechtigungen für die Amazon-S3-Buckets, in denen Canaries Testlaufergebnisse und andere Artefakte wie Protokolle und Screenshots speichern.

Achten Sie außerdem auf strenge Berechtigungen für die Speicherorte, an denen Ihr Canary-Quellcode gespeichert ist, so dass kein Benutzer versehentlich oder böswillig die Lambda-Ebenen oder Lambda-Funktionen löscht, die für das Canary verwendet werden.

Um sicherzustellen, dass Sie den beabsichtigten Canary-Code ausführen, können Sie das Objekt-Versioning für den Amazon-S3-Bucket verwenden, in dem Ihr Canary-Code gespeichert ist. Wenn Sie dann diesen Code angeben, um als Canary ausgeführt zu werden, können Sie das Objekt `versionId` als Teil des Pfades wie in den folgenden Beispielen einschließen.

```
https://bucket.s3.amazonaws.com/path/object.zip?versionId=version-id
```

```
https://s3.amazonaws.com/bucket/path/object.zip?versionId=version-id
https://bucket.s3-region.amazonaws.com/path/object.zip?versionId=version-id
```

Stack-Ablaufverfolgungen und Ausnahmemeldungen

Standardmäßig erfassen CloudWatch Synthetics Canaries jede Ausnahme, die von Ihrem Canary-Skript ausgelöst wird, unabhängig davon, ob das Skript benutzerdefiniert ist oder aus einem Blueprint stammt. CloudWatch Synthetics protokolliert sowohl die Ausnahmemeldung als auch den Stack-Trace an drei Orten:

- Zurück zum CloudWatch Synthetics-Service, um das Debuggen zu beschleunigen, wenn Sie Testläufe beschreiben
- In CloudWatch Logs entsprechend der Konfiguration, mit der Ihre Lambda-Funktionen erstellt wurden
- In die Synthetics-Protokolldatei, bei der es sich um eine Klartextdatei handelt, die in den Amazon-S3-Speicherort hochgeladen wird, der durch den Wert angegeben wird, den Sie für den `resultsLocation` des Canarys festgelegt haben

Wenn Sie weniger Informationen senden und speichern möchten, können Sie Ausnahmen erfassen, bevor sie in die CloudWatch Synthetics-Wrapper-Bibliothek zurückkehren.

Sie können in Ihren Fehlern auch Anforderungs-URLs angeben. CloudWatch Synthetics sucht in dem von Ihrem Skript ausgelösten Fehler nach URLs und entfernt je nach Konfiguration eingeschränkte URL-Parameter daraus. restrictedUrlParameters Wenn Sie in Ihrem Skript Fehlermeldungen protokollieren, können Sie mit getSanitizedErrorNachricht URLs vor der Protokollierung schwärzen.

Präzises Definieren des Geltungsbereichs Ihrer IAM-Rollen

Es wird empfohlen, Ihr Canary nicht so zu konfigurieren, dass potenziell bösartige URLs oder Endpunkte besucht werden. Wenn Sie Ihr Canary auf nicht vertrauenswürdige oder unbekannte Websites oder Endpunkte verweisen, kann Ihr Lambda-Funktionscode den Skripten bösartiger Benutzer zur Verfügung gestellt werden. Unter der Annahme, dass eine bösartige Website aus Chromium ausbrechen kann, könnte sie auf ähnliche Weise Zugriff auf Ihren Lambda-Code haben, wie wenn Sie mithilfe eines Internetbrowsers verbunden sind.

Führen Sie Ihre Lambda-Funktion mit einer IAM-Ausführungsrolle aus, die über eingeschränkte Berechtigungen verfügt. Wenn deine Lambda-Funktion durch ein bösartiges Skript beeinträchtigt

wird, sind die Aktionen, die sie ausführen kann, eingeschränkt, wenn sie als dein Canary-Konto ausgeführt wird. AWS

Wenn Sie die CloudWatch Konsole verwenden, um einen Canary zu erstellen, wird dieser mit einer speziellen IAM-Ausführungsrolle erstellt.

Schwärzung sensibler Daten

CloudWatch Synthetics erfasst URLs, Statuscode, Fehlerursache (falls vorhanden) sowie Header und Hauptteile von Anfragen und Antworten. Dies ermöglicht es einem Canary-Benutzer, Canarys zu verstehen, zu überwachen und zu debuggen.

Die Konfigurationen, die in den folgenden Abschnitten beschrieben werden, können an jedem Punkt in der Canary-Ausführung festgelegt werden. Sie können auch verschiedene Konfigurationen auf verschiedene Synthetics-Schritte anwenden.

Anfrage-URLs

Standardmäßig fordern CloudWatch Synthetics-Logs URLs, Statuscodes und den Statusgrund für jede URL in Canary-Logs an. Anforderungs-URLs können auch in Canary-Ausführungsberichten, HAR-Dateien usw. angezeigt werden. Ihre Anforderungs-URL kann sensible Abfrageparameter wie Zugriffstoken oder Passwörter enthalten. Sie können verhindern, dass vertrauliche Informationen von CloudWatch Synthetics protokolliert werden.

Um vertrauliche Informationen zu redigieren, legen Sie die Konfigurationseigenschaft fest. restrictedUrlParameters Weitere Informationen finden Sie unter SyntheticsConfiguration Klasse. Dies veranlasst CloudWatch Synthetics, URL-Parameter, einschließlich Pfad- und Abfrageparameterwerte, auf der Grundlage restrictedUrlParametersvor der Protokollierung zu redigieren. Wenn Sie in Ihrem Skript URLs protokollieren, können Sie mit getSanitizedUrl(url, stepConfig = null) URLs vor der Protokollierung schwärzen. Weitere Informationen finden Sie unter SyntheticsLogHelper Klasse.

Überschriften

Standardmäßig protokolliert CloudWatch Synthetics keine Anforderungs-/Antwort-Header. Bei UI-Canarys ist dies das Standardverhalten für Canarys, die Laufzeitversion `syn-nodejs-puppeteer-3.2` und höher verwenden.

Wenn Ihre Header keine vertraulichen Informationen enthalten, können Sie Header in HAR-Datei- und HTTP-Berichten aktivieren, indem Sie die Eigenschaften und auf setzen. includeRequestHeadersincludeResponseHeaders`true` Sie können alle Header aktivieren, aber die

Werte sensibler Header-Schlüssel einschränken. Sie können beispielsweise festlegen, dass nur `Authorization`-Header von Artefakten schwärzen, die von Canarys erzeugt wurden.

Anfrage- und Antworttext

Standardmäßig protokolliert CloudWatch Synthetics den Anforderungs-/Antworttext nicht in Canary-Protokollen oder Berichten. Diese Informationen sind besonders nützlich für API-Canarys. Synthetics erfasst alle HTTP-Anforderungen und kann Header, Anforderungs- und Antwortkörper anzeigen. Weitere Informationen finden Sie unter executeHttpStep(stepName, RequestOptions, [Rückruf], [StepConfig]). Sie können wählen, ob der Anforderungs-/Antworttext aktiviert werden soll, indem Sie die Eigenschaften und auf setzen. includeRequestBodyincludeResponseBody`true`

Protokollierung Amazon CloudWatch Amazon-API-Aufrufen mit AWS CloudTrail

Amazon CloudWatch und CloudWatch Synthetics sind in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem AWS Service ausgeführten Aktionen bereitstellt. CloudTrail erfasst API-Aufrufe, die von oder im Namen Ihres AWS Kontos getätigt wurden. Zu den erfassten Aufrufen gehören Aufrufe von der Konsole und Code-Aufrufe von API-Operationen.

Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen S3-Bucket aktivieren, einschließlich Ereignissen für CloudWatch. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, an die die Anfrage gestellt wurde CloudWatch, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und andere Details ermitteln.

Weitere Informationen darüber CloudTrail, einschließlich der Konfiguration und Aktivierung, finden Sie im AWS CloudTrail Benutzerhandbuch.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Root- oder AWS Identity and Access Management (IAM-) Benutzeranmeldedaten gestellt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie unter dem CloudTrail UserIdentity-Element.

Für eine fortlaufende Aufzeichnung der Ereignisse in deinem AWS Konto, einschließlich Events für CloudWatch und CloudWatch Synthetics, erstellst du einen Trail. Ein Trail ermöglicht die CloudTrail Übermittlung von Protokolldateien an einen S3-Bucket. Wenn Sie einen Trail in der Konsole erstellen, gilt der Trail standardmäßig für alle AWS Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen S3-Bucket. Sie können andere AWS Dienste so konfigurieren, dass sie die in den

CloudTrail Protokollen gesammelten Ereignisdaten weiter analysieren und darauf reagieren. Weitere Informationen finden Sie hier:

- Übersicht zum Erstellen eines Trails
- CloudTrail Unterstützte Dienste und Integrationen
- Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

> **Note**
>
> Informationen zu CloudWatch Logs-API-Aufrufen, die angemeldet sind CloudTrail, finden Sie unter CloudWatch Protokollinformationen in CloudTrail.

Themen

- CloudWatch Informationen in CloudTrail
- CloudWatch Internetmonitor in CloudTrail
- CloudWatch Informationen zu Synthetics in CloudTrail

CloudWatch Informationen in CloudTrail

CloudWatch unterstützt die Protokollierung der folgenden Aktionen als Ereignisse in CloudTrail Protokolldateien:

- DeleteAlarms
- DeleteAnomalyDetector
- DeleteDashboards
- DescribeAlarmHistory
- DescribeAlarms
- DescribeAlarmsForMetric
- DescribeAnomalyDetectors
- DisableAlarmActions
- EnableAlarmActions

- GetDashboard
- ListDashboards
- PutAnomalyDetector
- PutDashboard
- PutMetricAlarm
- SetAlarmState

Beispiel: Einträge in CloudWatch Protokolldateien

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `PutMetricAlarm` Aktion demonstriert.


```
{
    "Records": [{
        "eventVersion": "1.01",
        "userIdentity": {
            "type": "Root",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:root",
            "accountId": "123456789012",
            "accessKeyId": "EXAMPLE_KEY_ID"
        },
        "eventTime": "2014-03-23T21:50:34Z",
        "eventSource": "monitoring.amazonaws.com",
        "eventName": "PutMetricAlarm",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-ruby2/2.0.0.rc4 ruby/1.9.3 x86_64-linux Seahorse/0.1.0",
        "requestParameters": {
            "threshold": 50.0,
            "period": 60,
            "metricName": "CloudTrail Test",
            "evaluationPeriods": 3,
            "comparisonOperator": "GreaterThanThreshold",
            "namespace": "AWS/CloudWatch",
            "alarmName": "CloudTrail Test Alarm",
            "statistic": "Sum"
        },
        "responseElements": null,
        "requestID": "29184022-b2d5-11e3-a63d-9b463e6d0ff0",
```

```
        "eventID": "b096d5b7-dcf2-4399-998b-5a53eca76a27"
    },
    ..additional entries
  ]
  }
```

Der folgende Eintrag in der Protokolldatei zeigt, dass ein Benutzer die `PutRule` Aktion CloudWatch Ereignisse aufgerufen hat.

```
{
        "eventVersion":"1.03",
        "userIdentity":{
           "type":"Root",
           "principalId":"123456789012",
           "arn":"arn:aws:iam::123456789012:root",
           "accountId":"123456789012",
           "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
           "sessionContext":{
              "attributes":{
                 "mfaAuthenticated":"false",
                 "creationDate":"2015-11-17T23:56:15Z"
              }
           }
        },
        "eventTime":"2015-11-18T00:11:28Z",
        "eventSource":"events.amazonaws.com",
        "eventName":"PutRule",
        "awsRegion":"us-east-1",
        "sourceIPAddress":"AWS Internal",
        "userAgent":"AWS CloudWatch Console",
        "requestParameters":{
           "description":"",
           "name":"cttest2",
           "state":"ENABLED",
           "eventPattern":"{\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance
 State-change Notification\"]}",
           "scheduleExpression":""
        },
        "responseElements":{
           "ruleArn":"arn:aws:events:us-east-1:123456789012:rule/cttest2"
        },
        "requestID":"e9caf887-8d88-11e5-a331-3332aa445952",
        "eventID":"49d14f36-6450-44a5-a501-b0fdcdfaeb98",
```

```
         "eventType":"AwsApiCall",
         "apiVersion":"2015-10-07",
         "recipientAccountId":"123456789012"
}
```

Der folgende Eintrag in der Protokolldatei zeigt, dass ein Benutzer die `CreateExportTask` Aktion „ CloudWatch Protokolle“ aufgerufen hat.

```
{
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/someuser",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "someuser"
        },
        "eventTime": "2016-02-08T06:35:14Z",
        "eventSource": "logs.amazonaws.com",
        "eventName": "CreateExportTask",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-ruby2/2.0.0.rc4 ruby/1.9.3 x86_64-linux Seahorse/0.1.0",
        "requestParameters": {
            "destination": "yourdestination",
            "logGroupName": "yourloggroup",
            "to": 123456789012,
            "from": 0,
            "taskName": "yourtask"
        },
        "responseElements": {
            "taskId": "15e5e534-9548-44ab-a221-64d9d2b27b9b"
        },
        "requestID": "1cd74c1c-ce2e-12e6-99a9-8dbb26bd06c9",
        "eventID": "fd072859-bd7c-4865-9e76-8e364e89307c",
        "eventType": "AwsApiCall",
        "apiVersion": "20140328",
        "recipientAccountId": "123456789012"
}
```

CloudWatch Internetmonitor in CloudTrail

CloudWatch Internet Monitor unterstützt die Protokollierung der folgenden Aktionen als Ereignisse in CloudTrail Protokolldateien.

- CreateMonitor
- DeleteMonitor
- GetHealthEvent
- GetMonitor
- GetQueryResults
- GetQueryStatus
- ListHealthEvents
- ListMonitors
- ListTagsForResource
- StartQuery
- StopQuery
- UpdateMonitor

Beispiel: Einträge in der CloudWatch Internet Monitor-Protokolldatei

Das folgende Beispiel zeigt einen CloudTrail Internet Monitor-Protokolleintrag, der die `ListMonitors` Aktion veranschaulicht.

```
{
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::000000000000:assumed-role/role_name",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::000000000000:role/Admin",
                "accountId":"123456789012",
```

```
                "userName": "SAMPLE_NAME"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2022-10-11T17:25:41Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2022-10-11T17:30:18Z",
        "eventSource": "internetmonitor.amazonaws.com",
        "eventName": "ListMonitors",
        "awsRegion": "us-east-2",
        "sourceIPAddress": "192.0.2.0",
        "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)",
        "requestParameters": null,
        "responseElements": null,
        "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
        "readOnly": true,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "111122223333",
        "eventCategory": "Management"
    }
```

Das folgende Beispiel zeigt einen CloudTrail Internet Monitor-Protokolleintrag, der die `CreateMonitor` Aktion demonstriert.

```
{
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::000000000000:assumed-role/role_name",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::000000000000:role/Admin",
                "accountId":"123456789012",
```

```
                "userName": "SAMPLE_NAME"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2022-10-11T17:25:41Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2022-10-11T17:30:08Z",
        "eventSource": "internetmonitor.amazonaws.com",
        "eventName": "CreateMonitor",
        "awsRegion": "us-east-2",
        "sourceIPAddress": "192.0.2.0",
        "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)",
        "requestParameters": {
            "MonitorName": "TestMonitor",
            "Resources": ["arn:aws:ec2:us-east-2:444455556666:vpc/vpc-febc0b95"],
            "ClientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
        },
        "responseElements": {
            "Arn": "arn:aws:internetmonitor:us-east-2:444455556666:monitor/ct-
onboarding-test",
            "Status": "PENDING"
        },
        "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "111122223333",
        "eventCategory": "Management"
    }
```


CloudWatch Informationen zu Synthetics in CloudTrail

CloudWatch Synthetics unterstützt die Protokollierung der folgenden Aktionen als Ereignisse in CloudTrail Protokolldateien:

- CreateCanary
- DeleteCanary
- DescribeCanaries

- DescribeCanariesLastRun
- DescribeRuntimeVersions
- GetCanary
- GetCanaryRuns
- ListTagsForResource
- StartCanary
- StopCanary
- TagResource
- UntagResource
- UpdateCanary

Beispiel: CloudWatch Synthetics-Logdateieinträge

Das folgende Beispiel zeigt einen CloudTrail Synthetics-Protokolleintrag, der die `DescribeCanaries` Aktion demonstriert.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:assumed-role/role_name",
        "accountId":"123456789012",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111222333444:role/Administrator",
                "accountId":"123456789012",
                "userName": "SAMPLE_NAME"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-04-08T21:43:24Z"
            }
        }
```

```
    },
    "eventTime": "2020-04-08T23:06:47Z",
    "eventSource": "synthetics.amazonaws.com",
    "eventName": "DescribeCanaries",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.590
 Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.212-b03
 java/1.8.0_212 vendor/Oracle_Corporation",
    "requestParameters": null,
    "responseElements": null,
    "requestID": "201ed5f3-15db-4f87-94a4-123456789",
    "eventID": "73ddbd81-3dd0-4ada-b246-123456789",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Das folgende Beispiel zeigt einen CloudTrail Synthetics-Protokolleintrag, der die `UpdateCanary` Aktion demonstriert.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:assumed-role/role_name",
        "accountId":"123456789012",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
               "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111222333444:role/Administrator",
       "accountId":"123456789012",
                "userName": "SAMPLE_NAME"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-04-08T21:43:24Z"
            }
        }
```

```
    },
    "eventTime": "2020-04-08T23:06:47Z",
    "eventSource": "synthetics.amazonaws.com",
    "eventName": "UpdateCanary",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.590
 Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.212-b03
 java/1.8.0_212 vendor/Oracle_Corporation",
    "requestParameters": {
        "Schedule": {
            "Expression": "rate(1 minute)"
        },
        "name": "sample_canary_name",
        "Code": {
            "Handler": "myOwnScript.handler",
            "ZipFile": "SAMPLE_ZIP_FILE"
        }
    },
    "responseElements": null,
    "requestID": "fe4759b0-0849-4e0e-be71-1234567890",
    "eventID": "9dc60c83-c3c8-4fa5-bd02-1234567890",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Das folgende Beispiel zeigt einen CloudTrail Synthetics-Protokolleintrag, der die `GetCanaryRuns` Aktion demonstriert.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:assumed-role/role_name",
        "accountId":"123456789012",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111222333444:role/Administrator",
```

```
       "accountId":"123456789012",
                "userName": "SAMPLE_NAME"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-04-08T21:43:24Z"
            }
        }
    },
    "eventTime": "2020-04-08T23:06:30Z",
    "eventSource": "synthetics.amazonaws.com",
    "eventName": "GetCanaryRuns",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.590
 Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.212-b03
 java/1.8.0_212 vendor/Oracle_Corporation",
    "requestParameters": {
        "Filter": "TIME_RANGE",
        "name": "sample_canary_name",
        "FilterValues": [
            "2020-04-08T23:00:00.000Z",
            "2020-04-08T23:10:00.000Z"
        ]
    },
    "responseElements": null,
    "requestID": "2f56318c-cfbd-4b60-9d93-1234567890",
    "eventID": "52723fd9-4a54-478c-ac55-1234567890",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Verschlagworten Sie Ihre Amazon-Ressourcen CloudWatch

Ein Tag ist eine benutzerdefinierte Attributbezeichnung, die Sie oder einer AWS Ressource zuweisen. AWS Jedes Tag besteht aus zwei Teilen:

- einem Tag-Schlüssel (z. B. `CostCenter`, `Environment` oder `Project`). Bei Tag-Schlüsseln wird zwischen Groß- und Kleinschreibung unterschieden.
- einem optionalen Feld, dem sogenannten Tag-Wert (z. B. `111122223333` oder `Production`). Ein nicht angegebener Tag-Wert entspricht einer leeren Zeichenfolge. Wie bei Tag-Schlüsseln wird auch bei Tag-Werten zwischen Groß- und Kleinschreibung unterschieden.

Tags sind für folgende Aktivitäten nützlich:

- Identifizieren und organisieren Sie Ihre AWS Ressourcen. Viele AWS Dienste unterstützen Tagging, sodass Sie Ressourcen aus verschiedenen Diensten dasselbe Tag zuweisen können, um anzuzeigen, dass die Ressourcen miteinander verknüpft sind. Sie können beispielsweise einer CloudWatch Regel dasselbe Tag zuweisen, das Sie einer EC2-Instance zuweisen.

In den folgenden Abschnitten finden Sie weitere Informationen zu Tags für CloudWatch.

Unterstützte Ressourcen in CloudWatch

Die folgenden Ressourcen CloudWatch unterstützen Tagging:

- Alarme — Sie können Alarme mit dem AWS CLI Befehl tag-resource und der API kennzeichnen. TagResource Sie können Ihre Alarm-Tags auch auf der Seite mit den Alarmdetails in der CloudWatch Konsole anzeigen und verwalten.
- Kanarienvögel — Sie können Kanarienvögel mithilfe der CloudWatch Konsole taggen. Weitere Informationen finden Sie unter Erstellen eines Canarys.
- Contributor Insights-Regeln — Sie können Contributor Insights-Regeln bei der Erstellung taggen, indem Sie den put-insight-rule AWS CLI Befehl und die API verwenden. PutInsightRule Mithilfe des AWS CLI Befehls tag-resource und der API können Sie vorhandenen Regeln Tags hinzufügen. TagResource
- Metrik-Streams — Sie können Metrik-Streams taggen, wenn Sie sie erstellen, indem Sie den put-metric-stream AWS CLI Befehl und die PutMetricStreamAPI verwenden. Mithilfe des AWS CLI

Befehls tag-resource und der API können Sie vorhandenen Metrik-Streams Tags hinzufügen. TagResource

Weitere Informationen zum Hinzufügen und Verwalten von Tags finden Sie unter Verwalten von Tags.

Verwalten von Tags

Tags bestehen aus den Eigenschaften `Key` und `Value` für eine Ressource. Sie können die CloudWatch Konsole, die oder die CloudWatch API verwenden AWS CLI, um die Werte für diese Eigenschaften hinzuzufügen, zu bearbeiten oder zu löschen. Informationen zum Arbeiten mit Tags finden Sie unter:

- TagResourceUntagResource, und ListTagsForResourcein der Amazon CloudWatch API-Referenz
- tag-resource, untag-resource und list-tags-for-resourcein der Amazon CLI-Referenz CloudWatch
- Arbeiten mit dem Tag Editor im Ressourcengruppen-Benutzerhandbuch

Konventionen für die Tag-Benennung und -Verwendung

Die folgenden grundlegenden Benennungs- und Verwendungskonventionen gelten für die Verwendung von Tags mit Ressourcen: CloudWatch

- Jede Ressource kann maximal 50 Tags haben.
- Jeder Tag muss für jede Ressource eindeutig sein. Jeder Tag kann nur einen Wert haben.
- Die maximale Länge des Tag-Schlüssels beträgt 128 Unicode-Zeichen in UTF-8.
- Die maximale Länge des Tag-Wertes beträgt 256 Unicode-Zeichen in UTF-8.
- Erlaubte Zeichen sind Buchstaben, Ziffern und Leerzeichen, die in UTF-8 darstellbar sind, sowie die folgenden Zeichen: . : + = @ _ / - (Bindestrich).
- Bei Tag-Schlüsseln und -Werten muss die Groß- und Kleinschreibung beachtet werden. Eine bewährte Methode besteht darin, sich für eine einheitliche Schreibweise der Tag-Benennungen zu entscheiden und diese Strategie für alle Ressourcentypen umzusetzen. Entscheiden Sie sich beispielsweise für `Costcenter`, `costcenter` oder `CostCenter` und verwenden Sie diese Konvention für alle Tags. Vermeiden Sie die Verwendung von ähnlichen Tags mit uneinheitlicher Fallunterscheidung.

- Das `aws:` Präfix ist für Tags verboten, da es für die AWS Verwendung reserviert ist. Sie können keine Tag-Schlüssel oder -Werte mit diesem Präfix bearbeiten oder löschen. Tags mit diesem Präfix werden nicht zum Limit für Tags pro Ressource gezählt.

Grafana-Integration

Sie können Grafana-Version 6.5.0 und höher verwenden, um kontextuell durch die CloudWatch Konsole zu wechseln und mithilfe von Platzhaltern eine dynamische Liste von Metriken abzufragen. Auf diese Weise können Sie Metriken für AWS -Ressourcen wie Instances oder Container von Amazon Elastic Compute Cloud überwachen. Wenn neue Instances als Teil eines Auto Scaling-Ereignisses erstellt werden, werden sie automatisch im Diagramm angezeigt. Sie brauchen die neuen Instance-IDs nicht nachzuverfolgen. Vorgefertigte Dashboards erleichtern den Einstieg in die Überwachung von Amazon EC2, Amazon Elastic Block Store und Ressourcen. AWS Lambda

Sie können Grafana Version 7.0 und höher verwenden, um CloudWatch Logs Insights-Abfragen für Protokollgruppen in CloudWatch Logs durchzuführen. Sie können die Abfrageergebnisse in Balken-, Linien- und gestapelten Diagrammen sowie in einem Tabellenformat visualisieren. Weitere Informationen zu CloudWatch Logs Insights finden Sie unter Analysieren von Protokolldaten mit CloudWatch Logs Insights.

Weitere Informationen zu den ersten Schritten finden Sie unter Using AWS CloudWatch in Grafana in der Grafana Labs-Dokumentation.

Kontoübergreifende, regionsübergreifende Konsole CloudWatch

Wir empfehlen Ihnen, die kontenübergreifende Observability zu verwenden, um Ihre Messwerte, Logs und Traces bestmöglich zu nutzen. CloudWatch Weitere Informationen finden Sie unter CloudWatch kontenübergreifende Beobachtbarkeit.

CloudWatch bietet auch ein konto- und regionsübergreifendes Dashboard. CloudWatch Diese Funktionalität bietet Ihnen eine konto- und regionsübergreifende Transparenz für Ihre Dashboards, Alarme, Metriken und automatischen Dashboards. Es bietet keine kontoübergreifende Sichtbarkeit von Protokollen oder Ablaufverfolgungen.

Wenn Sie auch CloudWatch kontenübergreifende Observability verwenden, besteht ein Anwendungsfall für dieses kontenübergreifende CloudWatch Dashboard darin, dass eines Ihrer kontenübergreifenden Observability-Quellkonten die CloudWatch Metriken eines anderen Quellkontos sehen kann.

Im Rest dieses Abschnitts wird das konto- und regionsübergreifende Dashboard beschrieben. Sie können damit Dashboards erstellen, die CloudWatch Daten aus mehreren AWS Konten und mehreren Regionen in einem einzigen Dashboard zusammenfassen. AWS Sie können auch einen Alarm in einem Konto erstellen, das eine Metrik in einem anderen Konto überwacht.

Viele Organisationen haben ihre AWS Ressourcen in mehreren Konten bereitgestellt, um Abrechnungs- und Sicherheitsgrenzen festzulegen. In diesem Fall sollten Sie mindestens ein Konto als Überwachungskonto festlegen und in diesem Konto oder in diesen Konten Ihre regionsübergreifenden Dashboards erstellen.

Die kontenübergreifende Funktionalität ist integriert AWS Organizations, sodass Sie Ihre kontoübergreifenden Dashboards effizient erstellen können.

Regionsübergreifende Funktionalität

Die regionsübergreifende Funktionalität ist jetzt automatisch integriert. Sie müssen keine zusätzlichen Schritte ausführen, um Metriken aus verschiedenen Regionen in einem einzigen Konto im selben Diagramm oder Dashboard anzeigen zu können. Alarme werden nicht regionsübergreifend unterstützt, sodass Sie keinen Alarm in einer Region erstellen können, der eine Metrik in einer anderen Region überwacht.

Themen

Aktivierung kontenübergreifender Funktionen in CloudWatch

Um die kontoübergreifende Funktionalität in Ihrer CloudWatch Konsole einzurichten, verwenden Sie die CloudWatch Konsole, um Ihre Sharing-Konten und Monitoring-Konten einzurichten.

Ein Freigabekonto einrichten

Sie müssen die Freigabe in jedem Konto aktivieren, das dem Überwachungskonto Daten zur Verfügung stellt.

Dadurch werden die schreibgeschützten Berechtigungen, die Sie in Schritt 5 ausgewählt haben, allen Benutzern gewährt, die ein kontoübergreifendes Dashboard in dem Konto anzeigen, mit dem Sie teilen, wenn der Benutzer über entsprechende Berechtigungen in dem Konto verfügt, mit dem Sie teilen.

Damit dein Konto CloudWatch Daten mit anderen Konten teilen kann

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Settings (Einstellungen).
3. Wählen Sie für Share your CloudWatch data die Option Configure aus.
4. Wählen Sie unter Sharing (Freigeben) die Option Specific Accounts (Bestimmte Konten) und geben Sie die IDs der Konten ein, für die Sie Daten freigeben möchten.

   Alle Konten, die Sie hier angeben, können die CloudWatch Daten Ihres Kontos einsehen. Geben Sie nur die IDs von Konten an, die Sie kennen und denen Sie vertrauen.
5. Geben Sie unter Permissions (Berechtigungen) mit einer der folgenden Optionen an, wie Ihre Daten freigegeben werden sollen:

   - Bieten Sie Lesezugriff auf Ihre CloudWatch Kennzahlen, Dashboards und Alarme. Diese Option ermöglicht es den Überwachungskonten, kontoübergreifende Dashboards zu erstellen, die Widgets enthalten, die Daten aus Ihrem Konto enthalten CloudWatch .

- Fügen Sie CloudWatch automatische Dashboards hinzu. Wenn Sie diese Option auswählen, können Benutzer im Überwachungskonto auch die Informationen in den automatischen Dashboards dieses Kontos anzeigen. Weitere Informationen finden Sie unter Erste Schritte mit Amazon CloudWatch.
- Schreibgeschützten X-Ray-Zugriff für die X-Ray Trace Map einschließen. Wenn Sie diese Option auswählen, können Benutzer im Überwachungskonto auch die X-Ray Trace Map und die X-Ray-Ablaufverfolgungsinformationen in diesem Konto anzeigen. Weitere Informationen finden Sie unter Verwenden der X-Ray Trace Map.
- Full read-only access to everything in your account (Vollständiger schreibgeschützter Zugriff auf alles in Ihrem Konto). Mit dieser Option können die Konten, die Sie zum Teilen verwenden, kontoübergreifende Dashboards erstellen, die Widgets enthalten, die CloudWatch Daten aus Ihrem Konto enthalten. Außerdem können diese Konten tiefer in Ihr Konto sehen und die Daten Ihres Kontos in den Konsolen anderer AWS -Dienste anzeigen.

6. Wählen Sie Vorlage starten CloudFormation .

   Geben Sie auf dem Bestätigungsbildschirm **`Confirm`** ein und wählen Sie Launch template (Vorlage starten).

7. Aktivieren Sie das Kontrollkästchen I acknowledge... (Ich bestätige...). Wählen Sie anschließend Create stack (Stack erstellen) aus.

Freigeben für eine ganze Organisation

Wenn Sie das vorangegangene Verfahren ausführen, wird eine IAM-Rolle erstellt, mit der Ihr Konto Daten für ein Konto freigeben kann. Sie können eine IAM-Rolle erstellen oder bearbeiten, die Ihre Daten für alle Konten in einer Organisation freigibt. Tun Sie dies nur, wenn Sie alle Konten in der Organisation kennen und ihnen vertrauen.

Dadurch werden allen Benutzern, die ein kontoübergreifendes Dashboard in dem von Ihnen freigegebenen Konto anzeigen, die in den Richtlinien in Schritt 5 des vorherigen Verfahrens aufgeführten schreibgeschützten Berechtigungen erteilt, wenn der Benutzer über entsprechende Berechtigungen in dem von Ihnen freigegebenen Konto verfügt mit.

Um Ihre CloudWatch Kontodaten mit allen Konten in einer Organisation zu teilen

1. Falls Sie dies noch nicht getan haben, führen Sie das vorherige Verfahren aus, um Ihre Daten mit einem AWS Konto zu teilen.

2. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
3. Wählen Sie im Navigationsbereich Rollen aus.
4. Wählen Sie in der Rollenliste CloudWatch- CrossAccountSharingRole aus.
5. Wählen Sie auf der Registerkarte Trust Relationships (Vertrauensbeziehungen) Edit Trust Relationship (Vertrauensbeziehung bearbeiten) aus.

   Sie sehen eine Richtlinie wie diese:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

6. Ändern Sie die Richtlinie folgendermaßen, indem Sie *org-id* durch die ID Ihrer Organisation ersetzen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": "org-id"
        }
      }

    }
```

```
    ]
}
```

7. Wählen Sie Update Trust Policy (Trust Policy aktualisieren).

Einrichten eines Überwachungskontos

Aktivieren Sie jedes Überwachungskonto, wenn Sie kontoübergreifende CloudWatch Daten anzeigen möchten.

Wenn Sie das folgende Verfahren ausführen, CloudWatch wird eine dienstbezogene Rolle erstellt, die im Überwachungskonto auf Daten CloudWatch zugreift, die von Ihren anderen Konten gemeinsam genutzt wurden. Diese dienstverknüpfte Rolle heißt. AWSServiceRoleForCloudWatchCrossAccount Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für CloudWatch.

Um Ihrem Konto die Anzeige kontoübergreifender CloudWatch Daten zu ermöglichen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Einstellungen und dann im Abschnitt Kontoübergreifend und regionsübergreifend die Option Konfigurieren aus.
3. Wählen Sie im Abschnitt Kontoübergreifende regionsübergreifende Ansicht die Option Aktivieren aus und aktivieren Sie dann das Kontrollkästchen Auswahl in der Konsole anzeigen, damit eine Kontoauswahl in der CloudWatch Konsole angezeigt wird, wenn Sie eine Kennzahl grafisch darstellen oder einen Alarm erstellen.
4. Wählen Sie unter View cross-account cross-region (Konto- und regionenübergreifende Anzeige) eine der folgenden Optionen aus:

   - Account Id Input (Konto-ID-Eingabe). Mit dieser Option werden Sie aufgefordert, jedes Mal, wenn Sie Konten wechseln möchten, manuell eine Konto-ID einzugeben, wenn kontoübergreifende Daten angezeigt werden.
   - AWS Kontoauswahl für Unternehmen. Über diese Option können Sie die Konten anzeigen, die Sie während der kontoübergreifenden Integration mit Organizations angegeben haben. Wenn Sie die Konsole das nächste Mal verwenden, CloudWatch wird eine Dropdownliste mit diesen Konten angezeigt, aus der Sie auswählen können, wenn Sie kontoübergreifende Daten anzeigen.

Dazu müssen Sie zunächst Ihr Organisationsverwaltungskonto verwendet haben, um eine Liste der Konten in Ihrer Organisation anzeigen CloudWatch zu können. Weitere Informationen finden Sie unter (Optional) Integrieren Sie mit AWS Organizations.

- Custom account selector (Benutzerdefinierte Kontenauswahl). Diese Option fordert Sie auf, eine Liste mit Konto-IDs einzugeben. Wenn Sie die Konsole das nächste Mal verwenden, CloudWatch wird eine Dropdownliste mit diesen Konten angezeigt, aus der Sie auswählen können, wenn Sie kontenübergreifende Daten anzeigen.

  Sie können auch ein Label für jedes dieser Konten eingeben, um sie bei der Auswahl der anzuzeigenden Konten zu identifizieren.

  Die Kontoauswahleinstellungen, die ein Benutzer hier vornimmt, werden nur für diesen Benutzer beibehalten, nicht für die anderen Benutzer im Überwachungskonto.

5. Wählen Sie Enable (Aktivieren) aus.

Nach dem Abschluss dieser Einrichtung können Sie konto- und regionsübergreifende Dashboards erstellen. Weitere Informationen finden Sie unter Konto- und regionenübergreifende Dashboards.

(Optional) Integrieren Sie mit AWS Organizations

Wenn Sie kontenübergreifende Funktionen in integrieren möchten AWS Organizations, müssen Sie den Überwachungskonten eine Liste aller Konten in der Organisation zur Verfügung stellen.

Um die kontenübergreifende CloudWatch Funktionalität für den Zugriff auf eine Liste aller Konten in Ihrer Organisation zu aktivieren

1. Melden Sie sich beim Verwaltungskonto Ihrer Organisation an.
2. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
3. Wählen Sie im Navigationsbereich Settings (Einstellungen) und dann Configure (Konfigurieren) aus.
4. Wählen Sie unter Grant permission to view the list of accounts in the organization (Berechtigung zum Anzeigen der Liste der Konten in der Organisation erteilen) Specific accounts (Bestimmte Konten), um zur Eingabe einer Liste von Konto-IDs aufgefordert zu werden. Die Liste der Konten in Ihrer Organisation wird nur für die Konten freigegeben, die Sie hier angeben.
5. Wählen Sie Share organization account list (Organisationskontenliste freigeben).

6. Wählen Sie CloudFormation Vorlage starten.

   Geben Sie auf dem Bestätigungsbildschirm **`Confirm`** ein und wählen Sie Launch template (Vorlage starten).

Problembehebung bei Ihrer CloudWatch kontoübergreifenden Einrichtung

Dieser Abschnitt enthält Tipps zur Fehlerbehebung bei der kontoübergreifenden Konsolenbereitstellung in. CloudWatch

Ich erhalte Zugriffsverweigerungsfehler bei der Anzeige kontoübergreifender Daten

Überprüfen Sie, ob Folgendes der Fall ist:

- Ihr Monitoring-Konto sollte eine Rolle mit dem Namen AWSServiceRoleForCloudWatchCrossAccounthaben. Wenn dies nicht der Fall ist, müssen Sie diese Rolle erstellen. Weitere Informationen finden Sie unter Set Up a Monitoring Account.
- Jedes Sharing-Konto sollte eine Rolle mit dem Namen CloudWatch- habenCrossAccountSharingRole. Wenn dies nicht der Fall ist, müssen Sie diese Rolle erstellen. Weitere Informationen finden Sie unter Set Up A Sharing Account.
- Die Freigaberolle muss dem Überwachungskonto vertrauen.

Um zu überprüfen, ob Ihre Rollen für die CloudWatch kontoübergreifende Konsole ordnungsgemäß eingerichtet sind

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Rollen aus.
3. Stellen Sie in der Liste der Rollen sicher, dass die erforderliche Rolle vorhanden ist. Suchen Sie in einem Sharing-Konto nach CloudWatch- CrossAccountSharingRole. Suchen Sie in einem Überwachungskonto nach AWSServiceRoleForCloudWatchCrossAccount.
4. Wenn Sie ein Sharing-Konto haben und CloudWatch- CrossAccountSharingRole bereits vorhanden ist, wählen Sie CloudWatch- CrossAccountSharingRole.
5. Wählen Sie auf der Registerkarte Trust Relationships (Vertrauensbeziehungen) Edit Trust Relationship (Vertrauensbeziehung bearbeiten) aus.

6. Bestätigen Sie, dass die Richtlinie entweder die Konto-ID des Überwachungskontos oder die Organisations-ID einer Organisation enthält, die das Überwachungskonto enthält.

In der Konsole wird kein Konten-Dropdown-Menü angezeigt.

Überprüfen Sie zunächst, ob Sie die richtigen IAM-Rollen erstellt haben, wie im vorangegangenen Abschnitt zur Problembehandlung erläutert. Wenn diese korrekt eingerichtet sind, stellen Sie sicher, dass Sie dieses Konto aktiviert haben, um kontoübergreifende Daten anzuzeigen, wie unter Enable Your Account to View Cross-Account Data beschrieben.

Deaktivieren und Bereinigen nach kontoübergreifender Verwendung

Gehen Sie wie folgt vor, um die kontoübergreifende Funktionalität für CloudWatch zu deaktivieren.

Schritt 1: Kontoübergreifende Stacks oder Rollen entfernen

Die beste Methode besteht darin, die AWS CloudFormation Stacks zu entfernen, die zur Aktivierung der kontenübergreifenden Funktionalität verwendet wurden.

- Entfernen Sie in jedem der Sharing-Konten den Stapel CloudWatch- CrossAccountSharingRole.
- Wenn Sie früher AWS Organizations die kontoübergreifende Funktionalität für alle Konten in einer Organisation aktiviert haben, entfernen Sie den CrossAccountListAccountsRole Stapel CloudWatch- aus dem Verwaltungskonto der Organisation.

Wenn Sie die AWS CloudFormation Stacks nicht verwendet haben, um die kontenübergreifende Funktionalität zu aktivieren, gehen Sie wie folgt vor:

- Löschen Sie in jedem der Sharing-Konten die CloudWatchCrossAccountSharingRoleIAM-Rolle.
- Wenn Sie bisher AWS Organizations die kontoübergreifende Funktionalität für alle Konten in einer Organisation aktiviert haben, löschen Sie die ListAccountsRole IAM-Rolle CloudWatchCrossAccountSharing- - im Verwaltungskonto der Organisation.

Schritt 2: Serviceverknüpfte Rolle entfernen

Löschen Sie im Überwachungskonto die AWSServiceRoleForCloudWatchCrossAccountdienstverknüpfte IAM-Rolle.

CloudWatch Servicekontingenten

CloudWatch hat die folgenden Kontingente für Metriken, Alarme, API-Anfragen und Alarm-E-Mail-Benachrichtigungen.

> **Note**
>
> Für einige AWS Dienste CloudWatch, einschließlich, können Sie die CloudWatch Nutzungsmetriken verwenden, um Ihre aktuelle Servicenutzung in CloudWatch Diagrammen und Dashboards zu visualisieren. Sie können eine mathematische CloudWatch Metrikfunktion verwenden, um die Servicekontingente für diese Ressourcen in Ihren Diagrammen anzuzeigen. Sie können auch Alarme konfigurieren, die Sie warnen, wenn sich Ihre Nutzung einem Service Quotas nähert. Weitere Informationen finden Sie unter Visualisierung Ihrer Service Quotas und Einstellung von Alarmen.

Ressource	Standardkontingent
Alarmaktionen	5/Alarm. Dieses Kontingent kann nicht geändert werden.
Zeitraum für die Alarmauswertung	Der Maximalwert, der durch Multiplizieren des Alarmzeitraums mit der Anzahl der verwendeten Auswertungszeiträume berechnet wird, beträgt einen Tag (86 400 Sekunden). Dieses Kontingent kann nicht geändert werden.
Alarme	10/Monat/Kunde, kostenlos. Für zusätzliche Alarme fallen Gebühren an. Keine Begrenzung der Gesamtzahl der Alarme pro Konto. Alarme, die auf metrischen mathematischen Ausdrücken basieren, können bis zu 10 Metriken haben. 200 Metrics Insights-Alarme pro Region. Sie können eine Kontingenterhöhung beantragen.
Anomalieerkennungsmodelle	500 pro Region und Konto

Ressource	Standardkontingent
API-Anforderungen	1.000.000/Monat/Kunde, kostenlos.
Canarys	200 pro Region und Konto Sie können eine Kontingenterhöhung beantragen.
Contributor Insights-API-Anfragen	Die folgenden APIs haben eine Quote von 20 Transaktionen pro Sekunde (TPS) und pro Region. • DescribeInsightRules Dieses Kontingent kann nicht geändert werden. • GetInsightRuleReport Sie können eine Kontingenterhöhung beantragen. Die folgenden APIs haben eine Quote von 5 TPS pro Region. Dieses Kontingent kann nicht geändert werden. • DeleteInsightRules • PutInsightRule Die folgenden APIs besitzen ein Kontingent von 1 TPS pro Region. Dieses Kontingent kann nicht geändert werden. • DisableInsightRules • EnableInsightRules
Contributor Insights-Regeln	100 Regeln pro Region pro Konto. Sie können eine Kontingenterhöhung beantragen.
Benutzerdefinierte Metriken	Kein Kontingent.

Ressource	Standardkontingent
Dashboards	Bis zu 500 Widgets pro Dashboard. Bis zu 500 Metriken pro Dashboard-Widget. Bis zu 2500 Metriken pro Dashboard in allen Widgets. Diese Kontingente umfassen alle Metriken, die zur Verwendung in Metrikberechnungsfunktionen abgerufen werden, auch wenn diese Metriken nicht im Diagramm angezeigt werden. Diese Kontingente können nicht geändert werden.
DescribeAlarms	9 Transaktionen pro Sekunde (TPS) pro Region. Die maximale Anzahl der Operationsanforderungen, die Sie pro Sekunde ohne Drosselung senden können. Sie können eine Kontingenterhöhung beantragen.
DeleteAlarmsAnfrage DescribeAlarmHistoryAnfrage DisableAlarmActionsAnfrage EnableAlarmActionsAnfrage SetAlarmStateAnfrage	3 TPS pro Region für jede dieser Operationen. Die maximale Anzahl der Operationsanforderungen, die Sie pro Sekunde ohne Drosselung senden können. Diese Kontingente können nicht geändert werden.
DescribeAlarmsForMetricAnfrage	9 TPS pro Region. Die maximale Anzahl der Operation sanforderungen, die Sie pro Sekunde ohne Drosselung senden können. Dieses Kontingent kann nicht geändert werden.

Ressource	Standardkontingent
DeleteDashboardsAnfrage GetDashboardAnfrage ListDashboardsAnfrage PutDashboardAnfrage	10 TPS pro Region für jede dieser Operationen. Die maximale Anzahl der Operationsanforderungen, die Sie pro Sekunde ohne Drosselung senden können. Diese Kontingente können nicht geändert werden.
PutAnomalyDetector DescribeAnomalyDetectors	10 TPS pro Region. Die maximale Anzahl der Operation sanforderungen, die Sie pro Sekunde ohne Drosselung senden können.
DeleteAnomalyDetector	5 TPS pro Region. Die maximale Anzahl der Operation sanforderungen, die Sie pro Sekunde ohne Drosselung senden können.
Dimensionen	30/metric. Dieses Kontingent kann nicht geändert werden.

Ressource	Standardkontingent
GetMetricData	10 TPS pro Region für Vorgänge, die Metrics-Insights-Abfragen enthalten. Für Operationen, die keine Metrics-Insights-Abfragen enthalten, beträgt das Kontingent 50 TPS pro Region. Das ist die maximale Anzahl der Operationsanforderungen, die Sie pro Sekunde ohne Drosselung senden können. Sie können eine Kontingenterhöhung beantragen. Bei `GetMetricData` -Operationen, die eine Metrics-Insights-Abfrage enthalten, beträgt das Kontingent 4 300 000 Datenpunkte pro Sekunde (DPS) für die letzten 3 Stunden. Dies wird anhand der Gesamtzahl der von der Abfrage gescannten Datenpunkte berechnet (die nicht mehr als 10 000 Metriken enthalten können). 180.000 Datenpunkte pro Sekunde, wenn die in der API-Anforderung verwendete `StartTime` weniger oder genau drei Stunden vor dem aktuellen Zeitpunkt liegt. 396.000 Datenpunkte pro Sekunde, wenn die `StartTime` mehr als drei Stunden vor dem aktuellen Zeitpunkt liegt. Dies ist die maximale Anzahl an Datenpunkten, die Sie mithilfe mindestens eines oder mehrerer API-Aufrufe pro Sekunde ohne Drosselung anfordern können. Dieses Kontingent kann nicht geändert werden. Der DPS-Wert wird auf der Grundlage geschätzter und nicht der tatsächlichen Datenpunkte berechnet. Die Datenpunktschätzung wird auf der Grundlage des angeforderten Zeitbereichs, des Zeitraums und des Aufbewahrungszeitraums berechnet. Dies bedeutet: Wenn die tatsächlichen Datenpunkte in den angeforderten Metriken wenige oder null sind, tritt die Drosselung weiterhin auf, wenn die geschätzte Zahl der Datenpunkte

Ressource	Standardkontingent
	das Kontingent überschreitet. Das DPS-Kontingent ist pro Region.
GetMetricData	Ein einziger `GetMetricData` -Aufruf kann Folgendes beinhalten: • Bis zu 500 `MetricDataQuery` -Strukturen. • Bis zu 100 `SERVICE_QUOTA()` -Funktionen. • Bis zu 100 `SEARCH()`-Funktionen. • Bis zu 5 `LAMBDA()`-Funktionen. Diese Kontingente können nicht geändert werden.
GetMetricStatistics	400 TPS pro Region. Die maximale Anzahl der Operation sanforderungen, die Sie pro Sekunde ohne Drosselung senden können. Sie können eine Kontingenterhöhung beantragen.
GetMetricWidgetImage	Bis zu 500 Metriken pro Image. Dieses Kontingent kann nicht geändert werden. 20 TPS pro Region. Die maximale Anzahl der Operation sanforderungen, die Sie pro Sekunde ohne Drosselung senden können. Sie können eine Kontingenterhöhung beantragen.
ListMetrics	25 TPS pro Region. Die maximale Anzahl der Operation sanforderungen, die Sie pro Sekunde ohne Drosselung senden können. Sie können eine Kontingenterhöhung beantragen.

Ressource	Standardkontingent
Metrische Datenwerte	Der Wert eines metrischen Datenpunkts muss im Bereich von -2^360 bis 2^360 liegen. Spezielle Werte (z. B. NaN, +Infinity, -Infinity) werden nicht unterstützt. Dieses Kontingent kann nicht geändert werden.
MetricDatumArtikel	1000/ PutMetricDataAnfrage. Ein MetricDatumObjekt kann einen einzelnen Wert oder ein StatisticSetObjekt enthalten, das viele Werte repräsentiert. Dieses Kontingent kann nicht geändert werden.
Metriken	10/Monat/Kunde, kostenlos.
Metrics-Insights-Abfragen	Eine einzelne Abfrage kann nicht mehr als 10 000 Metriken verarbeiten. Dies bedeutet, dass, wenn die Klauseln SELECT, FROM und WHERE mehr als 10 000 Metriken entsprechen, verarbeitet die Abfrage nur die ersten 10 000 der gefundenen Metriken. Eine einzelne Abfrage kann nicht mehr als 500 Zeitreihen zurückgeben. Derzeit können Sie nur die letzten drei Stunden an Daten abfragen
API-Anforderungsraten für Observability Access Manager (OAM).	1 TPS pro Region für. PutSinkPolicy 10 TPS pro Region für jede andere CloudWatch OAM-API. Diese Kontingente spiegeln die maximale Anzahl von Operationsanfragen wider, die Sie pro Sekunde senden können, ohne eine Drosselung zu riskieren. Diese Kontingente können nicht geändert werden.

Ressource	Standardkontingent
Links zu OAM-Quellkonten	Jedes Quellkonto kann mit bis zu 5 Überwachungskonten verknüpft werden Dieses Kontingent kann nicht geändert werden.
OAM-Sinks	1 Senke pro Region und Konto Dieses Kontingent kann nicht geändert werden.
PutCompositeAlarmAnfrage	3 TPS pro Region. Die maximale Anzahl der Operation sanforderungen, die Sie pro Sekunde ohne Drosselung senden können. Sie können eine Kontingenterhöhung beantragen.
PutMetricAlarmAnfrage	3 TPS pro Region. Die maximale Anzahl der Operation sanforderungen, die Sie pro Sekunde ohne Drosselung senden können. Sie können eine Kontingenterhöhung beantragen.
PutMetricDataAnfrage	1 MB für HTTP-POST-Anfragen. PutMetricDatakann 500 Transaktionen pro Sekunde (TPS) verarbeiten. Dies ist die maximale Anzahl von Betriebsanforderungen, die Sie pro Sekunde stellen können, ohne dass sie gedrossel t werden. `PutMetricData` kann 1.000 Metriken pro Anfrage verarbeiten. Sie können eine Kontingenterhöhung beantragen.
Amazon-SNS-E-Mail-Benachric htigungen	1.000/Monat/Kunde, kostenlos.
Synthetics-Gruppen	20 pro Konto. Dieses Kontingent kann nicht geändert werden.

Ressource	Standardkontingent
TagResource	20 TPS pro Region. Die maximale Anzahl der Operation sanforderungen, die Sie pro Sekunde ohne Drosselung senden können. Dieses Kontingent kann nicht geändert werden.
UntagResource	20 TPS pro Region. Die maximale Anzahl der Operation sanforderungen, die Sie pro Sekunde ohne Drosselung senden können. Dieses Kontingent kann nicht geändert werden.

Dokumentverlauf

In der folgenden Tabelle werden wichtige Änderungen in den einzelnen Versionen des CloudWatch Amazon-Benutzerhandbuchs ab Juni 2018 beschrieben. Um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten, können Sie einen RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
CloudWatch Die Service Map von Application Signals unterstützt Canary- und RUM-Clients sowie Gruppen von AWS Serviceabhängigkeiten.	Die Vorschauversion von Application Signals hat der Service Map Standardg ruppierungen für Canaries, RUM-Clients und AWS Serviceabhängigkeiten desselben Typs hinzugefügt. Durch diese Änderung wird die Anzahl der Symbole in der Standardansicht der Service Map reduziert, um die Anzeige und Navigation zu erleichtern.	21. Mai 2024
CloudWatchReadOnlyAccess Die IAM-Richtlinie wurde aktualisiert	CloudWatch hat den Umfang einer Genehmigung in CloudWatchReadOnly Accessgeändert. Im Geltungsb ereich der Richtlinie wurden die `application-signals:List*` Aktionen `application-signal s:BatchGet*` `applicati on-signals:Get*` , und hinzugefügt, sodass Benutzer mithilfe von CloudWatc h Anwendungssignalen Probleme mit dem Zustand ihrer Dienste anzeigen,	17. Mai 2024

	untersuchen und diagnosti zieren können. CloudWatc hAußerdem wurde eine `iam:GetRole` Aktion hinzugefügt, mit der Benutzer überprüfen können, ob Application Signals eingerich tet ist.	
CloudWatchFullAccessDie V2-IAM-Richtlinie wurde aktualisi ert	CloudWatch hat den Umfang einer Genehmigung in CloudWatchFullAccessV2 geändert. Der Geltungsb ereich der Richtlinie wurde hinzugefügt, `application-signals:*` sodass Benutzer CloudWatch Anwendung ssignale verwenden können, um Probleme mit dem Zustand ihrer Dienste zu überprüfen, zu untersuchen und zu diagnosti zieren.	17. Mai 2024
Lambda Insights unterstützt AWS GovCloud (US-Ost) und AWS GovCloud (US-West)	CloudWatch Lambda Insights hat Unterstützung für die Regionen AWS GovCloud (USA-Ost) und AWS GovCloud (US-West) hinzugefügt.	29. April 2024

CloudWatch Die kontenübergreifende Observability unterstützt Ressourcenfilter	Sie können jetzt Filter erstellen, um anzugeben, welche Metrik-Namespaces und Protokollgruppen vom Quellkonto zum Überwachungskonto gemeinsam genutzt werden, wenn Sie die Verknüpfung zwischen den Konten erstellen.	26. April 2024
CloudWatch Aktualisierungen von Anwendungssignalen	Die Vorschauversion von Application Signals hat drei Funktionen hinzugefügt. Application Signals unterstützt jetzt Python-Anwendungen. Es bietet einen einfacheren Aktivierungsprozess für Anwendungen auf Amazon EKS-Architekturen. Und es enthält neue Konfigurationen, mit denen Sie die Kardinalität der gesammelten Metriken verwalten können.	26. April 2024
CloudWatch Container Insights mit verbesserter Observability für Amazon EKS kann AWS Elastic Fabric Adapter (EFA) -Metriken sammeln	Sie können jetzt CloudWatch Container Insights mit verbesserter Observability für Amazon EKS verwenden, um AWS Elastic Fabric Adapter (EFA) -Metriken aus Amazon EKS-Clustern zu sammeln.	23. April 2024

Die IAM-Richtlinie wurde aktualisiert	CloudWatch hat die CloudWatchApplicationSignalsServiceRolePolicyRichtlinie aktualisiert. Der Geltungsbereich der `logs:StartQuery` und der `logs:GetQueryResults` Berechtigungen in dieser Richtlinie wurde geändert, um Application Signals auf mehr Architekturen hinzuzufügen `arn:aws:logs:*:*:log-group:/aws/appsignals/*:*` und `"arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"` zu aktivieren. Diese Richtlinie ist der dienstbezogenen Rolle zugeordnet. AWSServiceRoleForCloudWatchApplicationSignals	18. April 2024
Internet Monitor bietet authentifizierten AWS Kunden eine globale Internet-Wetterkarte	Amazon CloudWatch Internet Monitor zeigt jetzt eine globale Internet-Wetterkarte an, die in der Konsole für alle authentifizierten AWS Kunden verfügbar ist. Um die Karte anzuzeigen, navigieren Sie in der CloudWatch Amazon-Konsole zu Internet Monitor.	16. April 2024

CloudWatch Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS kann AWS Neuron-Metriken sammeln	Sie können jetzt CloudWatch Container Insights mit verbesserter Observability für Amazon EKS verwenden, um AWS Neuron-Metriken aus Amazon EKS-Clustern zu sammeln.	16. April 2024
CloudWatch Application Signals fügt eine Registerkarte „Serviceübersicht“ und weitere Metriken zur Unterstützung der Diagnose hinzu	Auf der neuen Registerkarte „Serviceübersicht“ wird ein Überblick über Ihren Service angezeigt, einschließlich der Anzahl der Operationen, Abhängigkeiten, Synthetics und Client-Seiten. Auf der Registerkarte werden wichtige Kennzahlen für Ihren gesamten Service sowie die wichtigsten Abläufe und Abhängigkeiten angezeigt. Sie können jetzt auch Röntgenspuren anzeigen, die mit Problemen wie Fehlern, Fehlern und Latenzproblemen korrelieren.	16. April 2024
CloudWatch Container Insights mit verbesserter Observability für Amazon EKS bietet Unterstützung für Windows	Sie können jetzt CloudWatch Container Insights mit verbesserter Observability für Amazon EKS verwenden, um Metriken von Windows-Worker-Knoten auf Amazon EKS-Clustern zu sammeln.	10. April 2024

CloudWatchApplicationSignalsServiceRolePolicyDie IAM-Richtlinie wurde aktualisiert	CloudWatch hat den Umfang einer Genehmigung in CloudWatchApplicationSignalsServiceRolePolicy geändert. Der Umfang der `cloudwatch:GetMetricData` Genehmigung wurde dahingehend geändert, * dass Application Signals Messwerte aus Quellen verknüpfter Konten abrufen kann.	8. April 2024
Amazon CloudWatch Internet Monitor unterstützt jetzt kontoübergreifende Observability	Sie können jetzt die kontoübergreifende Beobachtbarkeit von Internet Monitor verwenden, um Ihre Anwendungen zu überwachen, die sich über mehrere AWS-Konten Anwendungen innerhalb einer einzigen Anwendung erstrecken. AWS-Region	29. März 2024

CloudWatchAgentServerPolicy und CloudWatchAgentAdminPolicydie Richtlinien wurden aktualisiert	CloudWatch `CloudWatchAgentServerPolicy` Sowohl den Richtlinien als auch den `CloudWatchAgentAdminPolicy` Richtlinien wurden Berechtigungen hinzugefügt, die es dem CloudWatch Agenten ermöglichen, X-Ray-Traces zu veröffentlichen und die Aufbewahrungsfristen für Protokollgruppen zu ändern. In beiden Richtlinien wurden die `logs:PutRetentionPolicy` Berechtigungen `xray:PutTraceSegments` `xray:PutTelemetryRecords` `xray:GetSamplingRules` ,`xray:GetSamplingTargets` ,, `xray:GetSamplingStatisticSummaries` und hinzugefügt	12. Februar 2024

Neue serviceverknüpfte Rolle und IAM-Richtlinie für CloudWatch Network Monitor	CloudWatch hat eine neue dienstbezogene Rolle mit dem Namen hinzugefügt. AWSServiceRoleForNetworkMonitor CloudWatch hat diese neue dienstbezogene Rolle hinzugefügt, damit Sie Monitore erstellen können, um Netzwerkmetriken zwischen Quellsubnetzen und Ziel-IP-Adressen abzurufen. Die neue CloudWatchNetworkMonitorServiceRolePolicyIAM-Richtlinie ist mit dieser Rolle verknüpft, und die Richtlinie erteilt die Erlaubnis, Netzwerkmetriken in CloudWatch Ihrem Namen abzurufen.	22. Dezember 2023
CloudWatch veröffentlicht Amazon CloudWatch Network Monitor	CloudWatch hat eine neue Funktion veröffentlicht, Amazon CloudWatch Network Monitor. Dies ist ein neuer aktiver Netzwerküberwachungsdienst, der feststellt, ob Netzwerkprobleme im AWS Netzwerk oder in Ihrem eigenen Unternehmensnetzwerk bestehen.	22. Dezember 2023

CloudWatchReadOnlyAccessDie Richtlinie wurde aktualisiert	CloudWatch Es wurden bestehende Nur-Lese-Berechtigungen für CloudWatch Synthetics, X-Ray und CloudWatch RUM sowie neue Nur-Lese-Berechtigungen für CloudWatch Application Signals hinzugefügt, CloudWatchReadOnlyAccesssodass Benutzer mit dieser Richtlinie die von Application Signals gemeldeten Dienstintegritätsprobleme untersuchen und diagnostizieren können. CloudWatch Die `cloudwatch:GenerateQuery` Berechtigung wurde hinzugefügt, damit Benutzer mit dieser Richtlinie aus einer Aufforderung in natürlicher Sprache eine CloudWatch Metrics Insights-Abfrageze ichenfolge generieren können.	05. Dezember 2023

CloudWatchFullAccessDie V2-Richtlinie wurde aktualisiert	CloudWatch hat bestehende Berechtigungen zu CloudWatchFullAccessV2 für CloudWatch Synthetics, X-Ray und CloudWatch RUM hinzugefügt und neue Berechtigungen für CloudWatch Application Signals hinzugefügt, sodass Benutzer mit dieser Richtlinie Application Signals vollständig verwalten können, um Probleme mit der Dienstintegrität zu analysieren und zu diagnostizieren.	05. Dezember 2023

Neue serviceverknüpfte Rolle und neue IAM-Richtlinie	CloudWatch hat eine neue dienstbezogene Rolle hinzugefügt, genannt. AWSServiceRoleForCloudWatchApplicationSignals CloudWatch hat diese neue dienstbezogene Rolle hinzugefügt, damit CloudWatch Application Signals CloudWatch Logdaten, X-Ray-Trace-Daten, CloudWatch Metrikdaten und Tagging-Daten von Anwendungen sammeln kann, die Sie für CloudWatch Application Signals aktiviert haben. Die neue CloudWatchApplicationSignalsServiceRolePolicy IAM-Richtlinie ist mit dieser Rolle verknüpft, und die Richtlinie erteilt CloudWatch Application Signals die Erlaubnis, Überwachungs- und Tagging-Daten von anderen relevanten Diensten zu sammeln. AWS	30. November 2023

CloudWatch veröffentlicht die Vorschauversion von Application Signals	CloudWatch Application Signals befindet sich in der Vorschauversion. Verwenden Sie Application Signals, um Ihre Anwendungen darauf AWS auszurichten, sodass Sie den aktuellen Zustand Ihrer Anwendungen überwachen, Service Level Objectives (SLOs) erstellen und die langfristige Anwendungsleistung anhand Ihrer Geschäftsziele verfolgen können. Weitere Informationen finden Sie unter Application Signals.	30. November 2023
CloudWatch fügt Unterstützung für die Abfrage anderer Datenquellen hinzu	Sie können CloudWatch es zum Abfragen, Visualisieren und Erstellen von Alarmen für Messwerte aus anderen Datenquellen verwenden. Weitere Informationen finden Sie unter Abfragen von Metriken aus anderen Datenquellen.	26. November 2023

CloudWatch Metrics Insights unterstützt die Generierung von Abfragen in natürlicher Sprache	CloudWatch Metrics Insights unterstützt Abfragen in natürlicher Sprache, um Abfragen zu generieren und zu aktualisieren. Weitere Informationen finden Sie unter Verwenden natürlicher Sprache zum Generieren und Aktualisieren von CloudWatch Metric Insights-Abfragen.	26. November 2023
CloudWatch veröffentlicht Container Insights mit verbesserter Beobachtbarkeit für Amazon EKS	CloudWatch hat eine neue Version von Container Insights veröffentlicht. Diese Version unterstützt eine verbesserte Beobachtbarkeit für Amazon-EKS-Cluster und kann detaillie rtere Metriken von Clustern sammeln, auf denen Amazon EKS ausgeführt wird. Nach der Installation sammelt sie automatisch detaillie rte Infrastrukturtelemetrie- und Container-Protokolle für Ihre Amazon-EKS-Cluster . Anschließend können Sie kuratierte, sofort verwendba re Dashboards nutzen, um die Anwendungs- und Infrastru kturtelemetrie genauer zu untersuchen.	6. November 2023

CloudWatch Metric Streams ermöglicht eine schnelle Partnereinrichtung	CloudWatch Metric Streams bietet jetzt eine schnelle Partner-Setup-Option, mit der Sie schnell einen Metrik-Stream für einige Drittanbieter einrichten können.	17. Oktober 2023
CloudWatch veröffentlicht Alarmempfehlungen	CloudWatch Synthetics bietet jetzt Alarmempfehlungen für Metriken aus anderen AWS Diensten. Mit diesen Empfehlungen können Sie die Metriken ermitteln, für die ein Alarm eingestellt werden sollte, um bewährte Methoden zur Überwachung dieser Services zu befolgen.	16. Oktober 2023
CloudWatch Synthetics veröffentlicht Runtime -6.0 syn-nodejs-puppeteer	CloudWatch Synthetics hat Runtime `syn-nodejs-puppeteer-6.0` veröffentlicht.	26. September 2023
Fügt Amazon CloudWatch Application Insights-Unterstützung für kontoübergreifende Anwendungen hinzu	Sie können jetzt CloudWatch Application Insights-Anwendungen über Kontogrenzen hinweg gemeinsam nutzen.	26. September 2023

Neue serviceverknüpfte Rolle und neue IAM-Richtlinie	CloudWatch hat eine neue dienstbezogene Rolle mit dem Namen AWSServiceRoleForCloudWatchMetrics_DbPerfInsightshinzugefügt. CloudWatch hat diese neue dienstbezogene Rolle hinzugefügt, um das Abrufen von Performance Insights Insights-Metriken für Alarme, Anomaliedetektion und Snapshots zu ermöglichen CloudWatch . Die neue AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicyIAM-Richtlinie ist mit dieser Rolle verknüpft, und die Richtlinie erteilt die Erlaubnis, Performance Insights Insights-Metriken in Ihrem Namen abzurufen. CloudWatch	20. September 2023
Fügt eine neue mathematische Metrikfunktion hinzu	CloudWatch hat eine neue mathematische Metrikfunktion hinzugefügt`DB_PERF_INSIGHTS` , mit der Sie Performance Insights Insights-Metriken aus AWS Datenbankdiensten abrufen können, um Alarme zu alarmieren, Anomalien zu erkennen und Schnappschüsse zu erstellen.	20. September 2023

CloudWatchReadOnlyAccessRichtlinie aktualisiert	CloudWatch hat die `application-autoscaling:DescribeScalingPolicies` Berechtigung hinzugefügt, CloudWatchReadOnlyAccesssodass Benutzer mit dieser Richtlinie auf Informationen über Application Auto Scaling Scaling-Richtlinien zugreifen können.	14. September 2023
CloudWatch Der Agent hat Unterstützung für AL2023 hinzugefügt	Der CloudWatch Agent unterstützt AL2023.	08. August 2023
Neue verwaltete IAM-Richtlinie, V2 CloudWatchFullAccess	CloudWatch hat eine neue Richtlinie CloudWatchFullAccessV2 hinzugefügt. Diese Richtlinie gewährt vollen Zugriff auf CloudWatch Aktionen und Ressourcen und begrenzt gleichzeitig den Umfang der Berechtigungen, die anderen Diensten wie Amazon SNS und gewährt wurden. Amazon EC2 Auto Scaling	1. August 2023

Aktualisierte serviceverknüpfte Rolle für Amazon CloudWatch Internet Monitor — Aktualisierung einer bestehenden Richtlinie	Fügt der serviceverknüpften Rolle für Internet Monitor neue Berechtigungen hinzu, `elasticloadbalancing:DescribeLoadBalancers` und `ec2:DescribeNetworkInterfaces`, um die Überwachung des Datenverkehrs für bestimmte Ressourcen von Network Load Balancer zu unterstützen.	25. Juli 2023
Unterstützung für Network Load Balancer Balancer-Ressourcen in Amazon CloudWatch Internet Monitor hinzugefügt	Fügt Unterstützung für die Erstellung eines Monitors in Internet Monitor mit spezifischen Ressourcen für Network Load Balancer hinzu, um detailliertere Ebenen der Beobachtbarkeit für Ihre Anwendung zu bieten.	25. Juli 2023

Feature für Dashboard-Variablen	CloudWatch veröffentlichte Dashboard-Variablen, mit denen Sie flexible Dashboards erstellen können, die schnell unterschiedliche Inhalte anzeigen können, je nachdem, wie Sie ein Eingabefeld im Dashboard eingerichtet haben. Sie können beispielsweise ein Dashboard erstellen, das schnell zwischen verschiedenen Lambda-Funktionen oder Amazon EC2 EC2-Instance-IDs wechseln kann, oder ein Dashboard, das zu verschiedenen AWS Regionen wechseln kann. Weitere Informationen finden Sie unter Flexible Dashboards mit Dashboard-Variablen erstellen.	28. Juni 2023
Internet Monitor unterstützt jetzt die Anpassung des Schwellenwerts für Zustandsereignisse	Internet Monitor hat die Möglichkeit hinzugefügt, den Schwellenwert für die Auslösung eines Zustandsereignisses durch einen globalen Leistungs- oder Verfügbarkeitswert anzupassen. Weitere Informationen finden Sie unter Leistung und Verfügbarkeit in Echtzeit verfolgen in Amazon CloudWatch Internet Monitor.	26. Juni 2023

Internet Monitor unterstützt jetzt alle kommerziellen Regionen	Internet Monitor hat sieben neue hinzugefügt AWS-Regionen und unterstützt jetzt alle kommerziellen Regionen.	19. Juni 2023
Neue Lambda-Insights-Erweiterungsversionen	CloudWatch Die Version 1.0.229.0 der Lambda Insights-Erweiterung wurde sowohl für x86-64-Plattformen als auch für ARM64-Plattformen hinzugefügt. Weitere Informationen finden Sie unter Verfügbare Versionen der Lambda-Insights-Erweiterung.	12. Juni 2023
CloudWatchReadOnlyAccessRichtlinie aktualisiert	CloudWatch Berechtigungen hinzugefügt zu CloudWatchReadOnlyAccess. Die `logs:StopLiveTail` Berechtigungen `logs:StartLiveTail` und wurden hinzugefügt, sodass Benutzer mit dieser Richtlinie die Konsole verwenden können, um CloudWatch Logs-Live-Tail-Sitzungen zu starten und zu beenden. Weitere Informationen finden Sie unter Use live tail to view logs in near real time.	6. Juni 2023

CloudWatch RUM bietet Unterstützung für benutzerdefinierte Metriken	Sie können CloudWatch RUM-App-Monitore verwenden, um benutzerdefinierte Metriken zu erstellen und diese an und CloudWatch CloudWatch Evidenly zu senden. Diese Funktion beinhaltet eine Aktualisierung der AmazonCloudWatchRUM-Richtlinie für ServiceRolePolicy verwaltetes IAM. In dieser Richtlinie wurde ein Bedingungsschlüssel geändert, sodass CloudWatch RUM benutzerdefinierte Metriken an benutzerdefinierte Metrik-Namespaces senden kann.	9. Februar 2023
Neue und aktualisierte verwaltete Richtlinien für CloudWatch	Um die CloudWatch kontenübergreifende Beobachtbarkeit zu unterstützen, wurden die `CloudWatchReadOnlyAccess` Richtlinien `CloudWatchFullAccess` und aktualisiert und die folgenden neuen verwalteten Richtlinien wurden hinzugefügt:`CloudWatchCrossAccountSharingConfiguration` , `OAMFullAccess` und. `OAMReadOnlyAccess` Weitere Informationen finden Sie unter CloudWatch Aktualisierungen der AWS verwalteten Richtlinien.	07. Februar 2023

CloudWatch Aktualisierungen der mit dem Dienst verknüpften Rollenrichtlinien von Application Insights — Aktualisierung einer bestehenden Richtlinie.	CloudWatch Application Insights hat eine bestehende Richtlinie für AWS dienstbezogene Rollen aktualisiert.	19. Dezember 2022
Amazon CloudWatch Application Insights-Unterstützung für containerisierte Anwendungen und Microservices über die Container Insights-Konsole.	Sie können die von CloudWatch Application Insights erkannten Probleme für Amazon ECS und Amazon EKS in Ihrem Container Insights-Dashboard anzeigen.	17. November 2021
Amazon CloudWatch Application Insights-Überwachung für SAP HANA-Datenbanken.	Sie können SAP HANA-Datenbanken mit Application Insights überwachen.	15. November 2021
Amazon CloudWatch Application Insights-Unterstützung für die Überwachung aller Ressourcen in einem Konto.	Sie können alle Ressourcen in einem Konto einbinden und überwachen.	15. September 2021
Amazon CloudWatch Application Insights-Unterstützung für Amazon FSx.	Sie können Metriken überwachen, die von Amazon FSx abgerufen wurden.	31. August 2021
SDK-Metriken werden nicht mehr unterstützt.	CloudWatch SDK Metrics wird nicht mehr unterstützt.	25. August 2021
Amazon CloudWatch Application Insights-Unterstützung für die Einrichtung der Container-Überwachung.	Sie können Container mithilfe von Best Practices mit Amazon CloudWatch Application Insights überwachen.	18. Mai 2021

Metrik-Streams sind allgemein verfügbar	Sie können Metrik-Streams verwenden, um CloudWatch Metriken kontinuierlich an ein Ziel Ihrer Wahl zu streamen. Weitere Informationen finden Sie unter Metric Streams im CloudWatch Amazon-Benutzerhandbuch.	31. März 2021
Amazon CloudWatch Application Insights-Überwachung für Oracle-Datenbanken auf Amazon RDS und Amazon EC2.	Mit Amazon CloudWatch Application Insights können Sie von Oracle abgerufene Metriken und Protokolle überwachen.	16. Januar 2021
Lambda Insights ist allgemein verfügbar	CloudWatch Lambda Insights ist eine Überwachungs- und Fehlerbehebungslösung für serverlose Anwendungen, die auf AWS Lambda ausgeführt werden. Weitere Informationen finden Sie unter Using Lambda Insights im CloudWatch Amazon-Benutzerhandbuch.	3. Dezember 2020
Amazon CloudWatch Application Insights-Überwachung für Prometheus JMX-Exportmetriken.	Sie können die vom Prometheus JMX-Exporter abgerufenen Metriken mit Amazon Application Insights überwachen. CloudWatch	20. November 2020

CloudWatch Synthetics veröffentlicht neue Runtime-Version	CloudWatch Synthetics hat eine neue Runtime-Version veröffentlicht. Weitere Informationen finden Sie unter Canary Runtime-Versionen im CloudWatch Amazon-Benutzerhandbuch.	11. September 2020
Amazon CloudWatch Application Insights-Überwachung für Postgre-SQL auf Amazon RDS und Amazon EC2.	Sie können Anwendungen überwachen, die mit PostgreSQL erstellt wurden, die auf Amazon RDS oder Amazon EC2 ausgeführt werden.	11. September 2020
CloudWatch unterstützt die gemeinsame Nutzung von Dashboards	Sie können CloudWatch Dashboards jetzt mit Personen außerhalb Ihrer Organisation und Ihres AWS Kontos teilen. Weitere Informationen finden Sie unter Sharing CloudWatch Dashboards im CloudWatch Amazon-Benutzerhandbuch.	10. September 2020
Richten Sie Monitore für .NET-Anwendungen mithilfe von SQL Server im Backend mit CloudWatch Application Insights ein	Sie können das Dokumentations-Tutorial verwenden, um Ihnen bei der Einrichtung von Monitoren für .NET-Anwendungen mithilfe von SQL Server im Backend mit CloudWatch Application Insights zu helfen.	19. August 2020

AWS CloudFormation Unterstützung für Amazon CloudWatch Application Insights-Anwendungen.	Sie können die CloudWatch Application Insights-Überwachung, einschließlich wichtiger Kennzahlen und Telemetrie, direkt aus AWS CloudFormation Vorlagen zu Ihrer Anwendung, Datenbank und Ihrem Webserver hinzufügen.	30. Juli 2020
Amazon CloudWatch Application Insights-Überwachung für Aurora für MySQL-Datenbankcluster.	Sie können Aurora für MySQL-Datenbankcluster (RDS Aurora) mit Amazon CloudWatch Application Insights überwachen.	2. Juli 2020
CloudWatch Allgemeine Verfügbarkeit von Contributor Insights	CloudWatch Contributor Insights ist jetzt allgemein verfügbar. Hiermit können Sie Protokolldaten analysieren und Zeitreihen erstellen, die Contributor-Daten anzeigen. Sie können Metriken über die Top-N-Contributors, die Gesamtzahl der eindeutigen Contributors und deren Nutzung anzeigen. Weitere Informationen finden Sie unter Verwenden von Contributor Insights zur Analyse von Daten mit hoher Kardinalität im Amazon-Benutzerhandbuch. CloudWatch	2. April 2020

CloudWatch Öffentliche Vorschau von Synthetics	CloudWatch Synthetics ist jetzt in der öffentlichen Vorschau. Hiermit können Sie Canaries erstellen, um Ihre Endpunkte und APIs zu überwachen. Weitere Informationen finden Sie unter Using Canaries im CloudWatch Amazon-Be nutzerhandbuch.	25. November 2019
CloudWatch Öffentliche Vorschau von Contributor Insights	CloudWatch Contribut or Insights ist jetzt in der öffentlichen Vorschauversion verfügbar. Hiermit können Sie Protokolldaten analysieren und Zeitreihen erstellen, die Contributor-Daten anzeigen. Sie können Metriken über die Top-N-Contributors, die Gesamtzahl der eindeutig en Contributors und deren Nutzung anzeigen. Weitere Informationen finden Sie unter Verwenden von Contribut or Insights zur Analyse von Daten mit hoher Kardinalität im Amazon-Benutzerhandbuch. CloudWatch	25. November 2019

CloudWatch startet die Funktion ServiceLens	ServiceLens verbessert die Beobachtbarkeit Ihrer Dienste und Anwendungen, indem Sie Traces, Metriken, Protokolle und Alarme an einem zentralen Ort integrieren können. ServiceLens integriert CloudWatch sich in AWS X-Ray , um einen end-to-end Überblick über Ihre Anwendung zu erhalten.	21. November 2019
Verwenden Sie CloudWatches, um Ihre AWS Servicekontingenten proaktiv zu verwalten	Sie können CloudWatch damit Ihre AWS Servicekontingenten proaktiv verwalten. CloudWatch Nutzungsmetriken bieten Einblick in die Nutzung von Ressourcen und API-Vorgängen durch Ihr Konto. Weitere Informationen finden Sie unter Service Quotas Integration and Usage Metrics im CloudWatch Amazon-Benutzerhandbuch.	19. November 2019
CloudWatch sendet Ereignisse, wenn sich der Status von Alarmen ändert	CloudWatch sendet jetzt ein Ereignis an Amazon, EventBridge wenn sich der Status eines CloudWatch Alarms ändert. Weitere Informationen finden Sie unter Alarmereignisse und EventBridge im CloudWatch Amazon-Benutzerhandbuch.	8. Oktober 2019

Container Insights	CloudWatch Container Insights ist jetzt allgemein verfügbar. Es ermöglicht Ihnen, Metriken und Protokolle aus Ihren containerisierten Anwendungen und Microservices zu sammeln, zu aggregieren und zusammenzufassen. Weitere Informationen finden Sie unter Using Container Insights im CloudWatch Amazon-Benutzerhandbuch.	30. August 2019
Updates für Container-Insights-Vorschaumetriken auf Amazon EKS und Kubernetes	Die öffentliche Vorversion von Container Insights auf Amazon EKS und Kubernetes wurde aktualisiert. InstanceId ist jetzt als Dimension in den Cluster-EC2-Instances enthalten. Auf diese Weise können Alarme, die für diese Metriken erstellt wurden, die folgenden EC2-Aktionen auslösen: Anhalten, Beenden, Neu Starten oder Wiederherstellen. Darüber hinaus werden Pod- und Service-Metriken jetzt vom Kubernetes-Namespace gemeldet, um die Überwachung und Alarme für Metriken nach Namespace zu vereinfachen.	19. August 2019

Updates für die Integration AWS Systems Manager OpsCenter	Updates zur Integration von CloudWatch Application Insights in Systems Manager OpsCenter.	7. August 2019
CloudWatch Nutzungsmetriken	CloudWatch Mithilfe von Nutzungsmetriken können Sie die Nutzung Ihrer CloudWatch Ressourcen verfolgen und Ihre Service-Limits einhalten. Weitere Informationen finden Sie unter https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch-Usage-Metrics.html.	6. August 2019
CloudWatch Öffentliche Vorschau von Container Insights	CloudWatch Container Insights ist jetzt in der öffentlichen Vorschauversion verfügbar. Es ermöglicht Ihnen, Metriken und Protokolle aus Ihren containerisierten Anwendungen und Microservices zu sammeln, zu aggregieren und zusammenzufassen. Weitere Informationen finden Sie unter Using Container Insights im CloudWatch Amazon-Benutzerhandbuch.	9. Juli 2019

CloudWatch Öffentliche Vorschau zur Erkennung von Anomalien	CloudWatch Die Anomaliee rkennung ist jetzt in der öffentlichen Vorschauv ersion verfügbar. CloudWatc h wendet Algorithmen für maschinelles Lernen auf die vergangenen Daten einer Metrik an, um ein Modell der erwarteten Werte der Metrik zu erstellen. Sie können dieses Modell zur Visualisi erung und zum Festlegen von Alarmen verwenden. Weitere Informationen finden Sie unter Verwenden der CloudWatc h Anomalieerkennung im CloudWatch Amazon-Be nutzerhandbuch.	9. Juli 2019
CloudWatch Einblicke in Anwendungen für.NET und SQL Server	CloudWatch Applicati on Insights für .NET und SQL Server erleichtert die Beobachtbarkeit von.NET- und SQL Server-Anwendungen . Es kann Ihnen helfen, die besten Überwachungen für Ihre Anwendungsressourc en einzurichten, um Daten kontinuierlich auf Anzeichen von Problemen mit Ihren Anwendungen zu analysieren.	21. Juni 2019

CloudWatch Der Agentenbereich wurde neu organisiert	Die CloudWatch Agentendokumentation wurde überarbeitet, um die Übersichtlichkeit zu verbessern, insbesondere für Kunden, die den Agenten über die Befehlszeile installieren und konfigurieren. Weitere Informationen finden Sie unter Erfassung von Metriken und Protokollen von Amazon EC2 EC2-Instances und lokalen Servern mit dem CloudWatch Agenten im CloudWatch Amazon-Benutzerhandbuch.	28. März 2019
Neue SEARCH-Funktion für Metrikberechnungen	In Metrikberechnungen steht jetzt die Funktion SEARCH zur Verfügung. Sie können damit Dashboards erstellen, die automatisch aktualisiert werden, sobald neue Ressourcen erstellt werden, die der Suchabfrage entsprechen. Weitere Informationen finden Sie unter Verwenden von Suchausdrücken in Diagrammen im CloudWatch Amazon-Benutzerhandbuch.	21. März 2019

AWS SDK-Metriken für Unternehmenssupport	Mit SDK Metrics können Sie den Zustand Ihrer AWS Dienste beurteilen und Latenzen diagnostizieren, die durch das Erreichen der Nutzungslimits Ihres Kontos oder durch einen Serviceausfall verursacht werden. Weitere Informationen finden Sie unter Überwachen von Anwendungen mithilfe von AWS SDK-Metriken im CloudWatch Amazon-Benutzerhandbuch.	11. Dezember 2018
Alarme bei mathematischen Ausdrücken	CloudWatch unterstützt die Erstellung von Alarmen auf der Grundlage metrischer mathematischer Ausdrücke. Weitere Informationen finden Sie unter Alarms on Math Expressions im CloudWatch Amazon-Benutzerhandbuch.	20. November 2018
Neue CloudWatch Konsolen-Startseite	Amazon hat in der CloudWatch Konsole eine neue Startseite erstellt, auf der automatisch wichtige Kennzahlen und Alarme für alle von Ihnen verwendeten AWS Dienste angezeigt werden. Weitere Informationen finden Sie unter Erste Schritte mit Amazon CloudWatch im CloudWatch Amazon-Benutzerhandbuch.	19. November 2018

AWS CloudFormation Vorlagen für den CloudWatch Agenten	Amazon hat AWS CloudFormation Vorlagen hochgeladen, mit denen Sie den CloudWatch Agenten installieren und aktualisieren können. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch unter AWS CloudFormation Installation des CloudWatch Agenten auf neuen Instances.	9. November 2018
Verbesserungen am CloudWatch Agenten	Der CloudWatch Agent wurde aktualisiert, sodass er sowohl mit den Protokollen StatsD als auch mit Collectd funktioniert. Außerdem verfügt er über eine verbesserte kontoüber greifende Unterstützung. Weitere Informationen finden Sie unter Abrufen benutzerdefinierter Metriken mit StatsD, Abrufen benutzerdefinierter Metriken mit collectd und Senden von Metriken und Protokollen an ein anderes AWS Konto im CloudWatch Amazon-Benutzerhandbuch.	28. September 2018

Unterstützung für Amazon VPC-Endpunkte	Sie können jetzt eine private Verbindung zwischen Ihrer VPC und CloudWatch herstellen. Weitere Informationen finden Sie unter Verwendung von VPC-Endpunkten CloudWatch mit Interface im CloudWatch Amazon-Benutzerhandbuch.	28. Juni 2018

In der folgenden Tabelle werden wichtige Änderungen am CloudWatch Amazon-Benutzerhandbuch vor Juni 2018 beschrieben.

Änderung	Beschreibung	Datum der Veröffentlichung
Metrikberechnungen	Sie können jetzt mathematische Ausdrücke auf CloudWatch Metriken anwenden und so neue Zeitreihen erstellen, die Sie den Diagrammen auf Ihrem Dashboard hinzufügen können. Weitere Informationen finden Sie unter Verwenden von Metrikberechnungen.	4. April 2018
"M von N"-Alarme	Sie können nun einen Alarm konfigurieren, der basierend auf "M von N"-Datenpunkten in einem beliebigen Auswertungsintervall ausgelöst wird. Weitere Informationen finden Sie unter Auswerten eines Alarms.	8. Dezember 2017
CloudWatch Agent	Ein neuer einheitlicher CloudWatch Agent wurde veröffentlicht. Sie können einen einzelnen einheitlichen Multiplattformagenten verwenden, um sowohl benutzerdefinierte Systemmetriken als auch ebensolche Protokolldateien von Amazon-EC2-Instances und On-Premises-Servern zu erfassen. Der neue Agent unterstützt sowohl Windows als	7. September 2017

Änderung	Beschreibung	Datum der Veröffentlichung
	auch Linux und ermöglicht die Anpassung der erfassten Metriken, einschließlich von Sub-Ressourcen-Metriken wie z. B. pro CPU-Kern. Weitere Informationen finden Sie unter Erfassen Sie mit dem CloudWatch Agenten Metriken, Logs und Traces.	
NAT-Gateway-Metriken	Es wurden Metriken für Amazon-VPC-NAT-Gateway hinzugefügt.	7. September 2017
Hochauflösende Metriken	Sie können jetzt optional benutzerdefinierte Metriken als hochauflösende Metriken, mit einer Granularität von bis zu einer Sekunde, einrichten. Weitere Informationen finden Sie unter Hochauflösende Metriken.	26. Juli 2017
Dashboard-APIs	Sie können jetzt Dashboards mithilfe von APIs und AWS CLI erstellen, ändern und löschen. Weitere Informationen finden Sie unter Ein CloudWatch Dashboard erstellen.	6. Juli 2017
AWS Direct Connect Metriken	Metriken für hinzugefügt AWS Direct Connect.	29. Juni 2017
Amazon-VPC-VPN-Metriken	Hinzugefügte Metriken für Amazon VPC VPN.	15. Mai 2017
AppStream 2.0-Metriken	Metriken für AppStream 2.0 hinzugefügt.	8. März 2017
CloudWatch Farbwähler für die Konsole	Sie können nun für jede Metrik auf Ihren Dashboard-Widgets die Farbe auswählen. Weitere Informationen finden Sie unter Bearbeiten Sie ein Diagramm auf einem Dashboard CloudWatch .	27. Februar 2017

Änderung	Beschreibung	Datum der Veröffentlichung
Alarme auf Dashboards	Dashboards können nun Alarme hinzugefügt werden. Weitere Informationen finden Sie unter Fügen Sie ein Alarm-Widget zu einem CloudWatch Dashboard hinzu oder entfernen Sie es .	15. Februar 2017
Metriken für Amazon Polly hinzugefügt	Metriken für Amazon Polly hinzugefügt.	1. Dezember 2016
Metriken für Amazon Managed Service für Apache Flink hinzugefügt	Metriken für Amazon Managed Service für Apache Flink hinzugefügt.	1. Dezember 2016
Zusätzlicher Support für Perzentil-Statistiken	Sie können ein beliebiges Perzentil mit bis zu zwei Dezimalstellen (z. B. p95,45) angeben. Weitere Informationen finden Sie unter Perzentile.	17. November 2016
Metriken für Amazon Simple Email Service hinzugefügt	Metriken für Amazon Simple Email Service hinzugefügt.	2. November 2016
Aktualisierte Aufbewahrung von Metriken	Amazon speichert Metrikdaten CloudWatch jetzt für 15 Monate statt für 14 Tage.	1. November 2016
Aktualisierte Konsolenschnittstelle für Metriken	Die CloudWatch Konsole wurde mit Verbesserungen vorhandener Funktionen und neuen Funktionen aktualisiert.	1. November 2016

Änderung	Beschreibung	Datum der Veröffentlichung
Metriken für Amazon Elastic Transcoder hinzugefügt	Metriken für Amazon Elastic Transcoder hinzugefügt.	20. September 2016
Metriken für Amazon API Gateway hinzugefügt	Metriken für Amazon API Gateway hinzugefügt.	9. September 2016
Metriken für hinzugefügt AWS Key Management Service	Metriken für hinzugefügt AWS Key Management Service.	9. September 2016
Metriken für die neuen von Elastic Load Balancing unterstützten Application Load Balancers wurden hinzugefügt	Hinzugefügte Metriken für Ihren Application Load Balancer	11. August 2016
Neue NetworkPacketsIn und NetworkPacketsOut Metriken für Amazon EC2 hinzugefügt	Neue NetworkPacketsIn und NetworkPacketsOut Metriken für Amazon EC2 hinzugefügt.	23. März 2016

Änderung	Beschreibung	Datum der Veröffentlichung
Hinzugefügte neue Metriken für Amazon-EC2-Spot-Flotte	Hinzugefügte neue Metriken für Amazon-EC2-Spot-Flotte.	21. März 2016
Neue CloudWatch Logs-Metriken hinzugefügt	Neue CloudWatch Logs-Metriken hinzugefügt.	10. März 2016
Amazon OpenSearch Service sowie AWS WAF Metriken und Dimensionen hinzugefügt	Amazon OpenSearch Service sowie AWS WAF Metriken und Dimensionen wurden hinzugefügt.	14. Oktober 2015
Unterstützung für CloudWatch Dashboards hinzugefügt	Dashboards sind anpassbare Homepages in der CloudWatch Konsole, mit denen Sie Ihre Ressourcen in einer einzigen Ansicht überwachen können, auch wenn sie über verschiedene Regionen verteilt sind. Weitere Informationen finden Sie unter CloudWatch Amazon-Dashboards verwenden.	8. Oktober 2015
AWS Lambda Metriken und Dimensionen hinzugefügt	AWS Lambda Metriken und Dimensionen hinzugefügt.	4. September 2015

Änderung	Beschreibung	Datum der Veröffentlichung
Hinzugefügte Metriken und Dimensionen für Amazon Elastic Container Service	Metriken und Dimensionen für den Amazon Elastic Container Service hinzugefügt.	17. August 2015
Metriken und Dimensionen von Amazon Simple Storage Service hinzugefügt	Metriken und Dimensionen für den Amazon Simple Storage Service hinzugefügt.	26. Juli 2015
Neues Feature: Neustarten einer Alarmaktion	Neustart-Alarmaktion und neue IAM-Rolle für die Verwendung mit Alarmaktionen hinzugefügt. Weitere Informationen finden Sie unter Erstellen Sie Alarme, um eine EC2-Instance anzuhalten, zu beenden, neu zu starten oder wiederherzustellen.	23. Juli 2015
WorkSpaces Amazon-Metriken und -Dimensionen hinzugefügt	WorkSpaces Amazon-Metriken und -Dimensionen hinzugefügt.	30. April 2015
Metriken und Dimensionen für Amazon Machine Learning hinzugefügt	Metriken und Dimensionen für Amazon Machine Learning hinzugefügt.	9. April 2015

Änderung	Beschreibung	Datum der Veröffentlichung
Neues Feature: Alarmaktionen für Amazon-EC2-Instance-Wiederherstellung	Aktualisierte Alarmaktionen, um eine neue EC2-Instance-Wiederherstellungsaktion aufzunehmen. Weitere Informationen finden Sie unter Erstellen Sie Alarme, um eine EC2-Instance anzuhalten, zu beenden, neu zu starten oder wiederherzustellen.	12. März 2015
Amazon CloudFront - und CloudSearch Amazon-Metriken und -Dimensionen hinzugefügt	Amazon CloudFront - und CloudSearch Amazon-Metriken und -Dimensionen hinzugefügt.	6. März 2015
Metriken und Dimensionen für Amazon Simple Workflow Service hinzugefügt	Metriken und Dimensionen für den Amazon Simple Workflow Service hinzugefügt.	9. Mai 2014
Aktualisierter Leitfaden zum Hinzufügen von Unterstützung für AWS CloudTrail	Es wurde ein neues Thema hinzugefügt, in dem erklärt wird AWS CloudTrail , wie Sie Aktivitäten in Amazon protokollieren können CloudWatch. Weitere Informationen finden Sie unter Protokollierung Amazon CloudWatch Amazon-API-Aufrufen mit AWS CloudTrail.	30. April 2014

Änderung	Beschreibung	Datum der Veröffentlichung
Die Anleitung zur Verwendung des neuen AWS Command Line Interface (AWS CLI) wurde aktualisiert	Die AWS CLI ist eine serviceübergreifende CLI mit einer vereinfachten Installation, einer einheitlichen Konfiguration und einer konsistenten Befehlsze ilensyntax. Die AWS CLI wird unter Linux/Unix, Windows und Mac unterstützt. Die CLI-Beispiele in diesem Handbuch wurden aktualisiert, um die neue AWS CLI zu verwenden. Informationen zur Installation und Konfiguration der neuen AWS CLI finden Sie unter Getting Up with the AWS CLI Interface im AWS Command Line Interface Benutzerhandbuch.	21. Februar 2014
Amazon Redshift sowie AWS OpsWorks Metriken und Dimensionen hinzugefügt	Amazon Redshift sowie AWS OpsWorks Metriken und Dimensionen wurden hinzugefügt.	16. Juli 2013
Metriken und Dimensionen von Amazon Route 53 hinzugefügt	Metriken und Dimensionen von Amazon Route 53 hinzugefügt.	26. Juni 2013
Neue Funktion: Amazon CloudWatch Alarm Actions	Es wurde ein neuer Abschnitt hinzugefügt, um CloudWatch Amazon-Alarmaktionen zu dokumenti eren, mit denen Sie eine Amazon Elastic Compute Cloud-Instance stoppen oder beenden können. Weitere Informationen finden Sie unter Erstellen Sie Alarme, um eine EC2-Instance anzuhalten, zu beenden, neu zu starten oder wiederherzustellen.	8. Januar 2013

Änderung	Beschreibung	Datum der Veröffentlichung
Aktualisierte EBS-Metriken	EBS-Metriken aktualisiert, um zwei neue Metriken für Volumes mit bereitgestellten IOPS aufzunehmen.	20. November 2012
Neue Gebührenlimit-Warnungen	Sie können jetzt Ihre AWS Gebühren anhand von CloudWatch Amazon-Metriken überwachen und Alarme einrichten, um Sie zu benachrichtigen, wenn Sie den angegebenen Schwellenwert überschritten haben. Weitere Informationen finden Sie unter Erstellen Sie einen Abrechnungsalarm, um Ihre geschätzten AWS Gebühren zu überwachen.	10. Mai 2012
Neue Metriken	Sie können nun auf sechs neue Metriken von Elastic Load Balancing zugreifen, die Zählungen der verschiedenen HTTP-Antwortcodes bereitstellen.	19. Oktober 2011
Neues Feature	Sie können nun auf Metriken aus Amazon EMR zugreifen.	30. Juni 2011
Neues Feature	Sie können jetzt über den Amazon Simple Notification Service und den Amazon Simple Queue Service auf Metriken zugreifen.	14. Juli 2011
Neues Feature	Zusätzliche Informationen zur Verwendung der API `PutMetricData` , um benutzerdefinierte Metriken zu veröffentlichen. Weitere Informationen finden Sie unter Veröffentlichen von benutzerdefinierten Metriken.	10. Mail 2011

Änderung	Beschreibung	Datum der Veröffentlichung
Aktualisierte Aufbewahrung von Metriken	Amazon speichert den Verlauf eines Alarms CloudWatch jetzt für zwei Wochen statt für sechs Wochen. Mit dieser Änderung entspricht der Aufbewahrungszeitraum für Alarme dem Aufbewahrungszeitraum für Metrikdaten.	7. April 2011
Neues Feature	Zusätzliche Möglichkeit, um Amazon Simple Notification Service oder Auto-Scaling-Benachrichtigungen zu senden, sobald eine Metrik einen Grenzwert überschritten hat. Weitere Informationen finden Sie unter Alarme.	2. Dezember 2010
Neues Feature	Zu einer Reihe von CloudWatch Aktionen gehören jetzt die NextToken Parameter MaxRecords und, mit denen Sie die Anzeige von Ergebnisseiten steuern können.	2. Dezember 2010
Neues Feature	Dieser Service ist jetzt in AWS Identity and Access Management (IAM) integriert.	2. Dezember 2010

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.